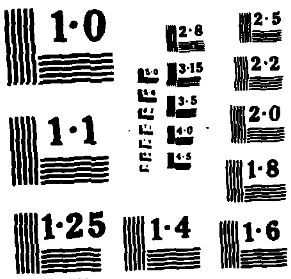
AD-A161 896 FLUTTER GENERATOR CONTROL AND FORCE COMPUTER(U) 1/1
ADVANCED ENGINEERING LAB ADELAIDE (AUSTRALIA)
R W LEVINGE JUL 85 AEL-0242-TM

UNCLASSIFIED F/G 1/3 NL

END
DATE
FILMED
1-86
DTIC


1·0
2·8
2·5
3·15
2·2
3·5
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6

(12)

AEL-0242-TM

AR-004-156

AUSTRALIA

AD-A161 896

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

ADVANCED ENGINEERING LABORATORY

DEFENCE RESEARCH CENTRE SALISBURY
SOUTH AUSTRALIA

TECHNICAL MEMORANDUM

AEL-0242-TM

FLUTTER GENERATOR CONTROL AND FORCE COMPUTER

R.W. LEVINGE

Technical Memoranda are of a tentative nature, representing the views of the author(s), and do not necessarily carry the authority of the Laboratory.

Approved for Public Release

COPY No. 25

UNCLASSIFIED

AR-004-156

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

ADVANCED ENGINEERING LABORATORY

TECHNICAL MEMORANDUM

AEL-0242-TM

FLUTTER GENERATOR CONTROL AND FORCE COMPUTER

R.W. Levinge

S U M M A R Y

It is required to investigate the possibility of flutter induced by a store carried under the wing of an aircraft. This involves in-flight dynamic analysis of structural deformations at given points on an airframe due to forces originating in the store. A system of rotating eccentric masses generates a force spectrum 2.4 to 20.0 Hz in both horizontal and vertical axes. Electronically controlled, the "Flutter Generator" runs for 28 s with a swept frequency and a peak force of 800 N. The vertical component of force is computed continuously and telemetered to ground as an analogue signal.

POSTAL ADDRESS: Director, Advanced Engineering Laboratory,
Box 2151, GPO, Adelaide, South Australia, 5001.

UNCLASSIFIED

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

The flutter generator and control circuits, contained in the store (figure 1), are powered from aircraft supplies: 400 Hz 115/208 V 3-phase and + 28 V dc. Additional regulated aircraft supplies give ±20 V and +9 V dc. A 2.5 hp motor drives the masses with a maximum 2 kW loading on the supplies. Controls and indicators are located in the cockpit.

Centrifugal forces of up to 800 N are generated by 4 masses rotating at different speeds about a common horizontal axis. The cyclic vertical component has a frequency equal to mass angular velocity (ω). Masses, initially aligned at bottom dead centre, are realigned at the end of a run. Controlled motor speed, increasing for about 14 s then decreasing for 14 s, gives a rate of change of frequency almost proportional to frequency. The force waveform is a dual sinewave sweeping from 2.48 to 7.05 Hz and from 7.05 to 20 Hz.

A computed voltage analagous to force, scaled 199 N/V for steel masses, is accurate to ±23 N. For aluminium masses scaling is 66 N/V ±7.5 N. Maximum frequency domain errors are ±5° phase and ±5% magnitude over the band 2.5 Hz to 20 Hz. Computing time and smoothing give an 850 µs transport lag. All errors are referred to the telemetry input.

The force compute routine, based on 16-bit numbers, is executed at least 600 times/s; average execution time being 460 µs. Thus quantisation errors are small and very little smoothing is required in the analogue output. Other routines calculate motor demand speed, check data, operate relays and indicators and protect against mechanical and signal failures. Speed control uses a separate analogue system.

A switch panel in the cockpit enables the pilot to power-up, run and abort the system. Power-on initialises the system, driving the motor slowly to align the masses. Pressing "run" starts the generator and, after about 28 s, it stops at the point of mass realignment, ready for the next run. Coloured lamps on the panel indicate system states.

2. SPEED CONTROL

The speed control loop is shown in figure 4 where signal voltages are referenced in the text. Transfer functions of the main blocks are non linear and analysis was made using a mathematical model.

Triacs connected to the three phase supply feed variable power to a series wound motor. An amplified dc error voltage, demand speed minus motor speed, sets the firing angles. A positive error produces power to accelerate the system while a negative error produces no power and friction decelerates the system. The motor runs in one direction, at or above the demand speed. Dynamic braking at the end of the run stops the system within one revolution of mass A.

Force computation is based on measured speed so small deviations from demand speed are unimportant.

Demand speed is to be increased during 13.85 s as follows:-

$$\omega_d = 404[1 + 1.85(0.0722t)^2] \quad \text{rad/s} \tag{1}$$

and decreased during the next 13.85 s as follows:-

$$\omega_d = 404[1 + 1.85(1 - 0.0722t)^3] \quad \text{rad/s} \qquad (2)$$

26:1 gearing to mass A, gives a range: 2.475 to 7.05 rev/s.

The control system derives all its power from the 400 Hz supply to which it presents a balanced load.

2.1 Motor characteristics

A Black and Decker 2.5 hp series-wound dc motor is characterised:-

$$T(6.5 + 0.0153\omega_o)^2 = 0.006\ V^2 \qquad (3)$$

where V = volts across motor

ω_o = motor angular velocity (rad/s)

T = motor torque (Nm)

Total Copper loss = 6.5 ohms

Over a speed range 404 to 930 rad/s, and a voltage of 250 V, more than 2.5 hp was developed. Maximum available voltage is 270 V. More than 2 hp was developed up to 1152 rad/s.

2.2 Mechanical load

The load (figure 2) comprises the 4 masses, A, B, C and D, driven about a horizontal axis. Gravity exerts torque on each mass proportional to Sin θ where (θ) is displacement from bottom dead centre. Mass Radius (MR) products, gear ratios and peak torques on the motor shaft are given for each mass in Table 1. To refer to the motor shaft, (θ) is multiplied by the gear ratio giving (θ_o).

TABLE 1. STEEL MASSES

	MR Product	Gear Ratio	Peak Torque
Mass A	0.3471 kg.m	26.00:1	0.131 N.m
Mass B	0.4460 kg.m	25.74:1	0.170 N.m
Mass C	0.0442 kg.m	9.03:1	0.048 N.m
Mass D	0.0565 kg.m	9.99:1	0.062 N.m

Aluminium masses give MR products and peak torques of about one third of those in Table 1. In flight, aircraft acceleration affects torques, altering the load. On the ground, total load torque is as follows:-

$$T = 0.00046\alpha_o - 0.131\text{Sin}(0.0385\theta_o) - 0.170\text{Sin}(0.0389\theta_o)$$
$$- 0.048\text{Sin}(0.11\theta_o - 0.062\text{Sin}(0.111\theta_o) + F \qquad (4)$$

where θ_0 = displacement of motor shaft from start (rad)

α_0 = acceleration (rad/s²)

F = friction (N)

"F" is the total coulomb friction of the system.

Energy to excite airframe flutter modes while carrying the store is an additional unknown load on the motor. However, under laboratory conditions, the motor uses only a fraction of its available 2.5 hp.

2.3 Motor drive amplifier

Output voltage from the rectifier/triac circuit (figure 3) is a function of the conduction angle of each triac. In a 400 Hz 3-phase system, maximum conduction angle is 120° and minimum ripple frequency is 2400 Hz.

Three triac controllers (Philips TCA 280A) with a common dc input generate the respective firing pulses. While conducting, voltage across any triac is low; its output equals input for either polarity. Assuming phase-1 triac conducts, phase-2 triac cannot be fired until phase 2 V exceed phase 1. Thus the firing angle (a) is never less than 30°. Furthermore, when phase 2 reaches 30°, phase 1 is 150° at which point phase-1 triac ceases to conduct. Maximum conduction angle for each of the triacs is therefore limited to between 30° and 150°. A train of pulses ensures firing when the initial pulse occurs before the 30° point.

The amplifier output, for firing angle "a" between 60° and 180°, is:

$$V_2 = 155.2[1 + Cos(a)] \quad V \tag{5}$$

and between 30° and 60°, is:

$$V_2 = 155.2[0.866Sin(a) + 1.5Cos(a)] \quad V \tag{6}$$

Equation (7) gives "a" as a function of servo amplifier output volts V_1.

$$a = 90(V_1 - 2.75) \quad \text{degrees} \tag{7}$$

Waveform distortion and radio frequency interference (rfi) on the 3-phase lines are filtered.

2.4 Motor velocity

A light-beam modulator on the motor shaft (figure 5), outputs 20 pulses/rev or between 1300 and 3700 pulses/s over a 404 to 1152 rad/s speed range. These pulse rates, converted to dc using an LM 2917N, give a measure of motor velocity scaled 1.92 mV/rad/s. A 2.6 to 7.4 kHz ripple is attenuated by an internal 2 ms lag (equation (8)).

$$\frac{V_s}{\theta_0} = \frac{0.00192}{1 + s0.002} \quad V/rad/s \tag{8}$$

The output amplifier, (equation (9)) raises the level to 5.2 mV/rad/s and attenuates ripple by at least 70 dB without destabilising control.

$$\frac{V_4}{V_5} = \frac{2.7}{1 + s0.0009 + s^2 0.00000032} \qquad (9)$$

Maximum output excursion is 4 V.

2.5 Servo amplifier

Motor speed: +5.2 mV/rad/s, added to demand speed: -12 mV/rad/s, via scaled summing resistors (R41 and R51 figure 5) gives the error. Positive errors (negative voltages) are amplified by a factor of 70 and inverted. The output for maximum positive error is +2.75 V and, for zero or any negative error, is +7 V. An output of 2.75 V, fed to the common input of the TCA 280A controllers, sets the firing angle of each triac to 30° giving the maximum 120° conduction angles. An output of 7 V sets firing angles to 180° giving zero conduction angles.

$$V_1 = -70(V_4 - 0.433.V_6) + 7 \qquad |V_4| < |0.433.V_6| \qquad (10)$$

$$V_1 = +7 \qquad |V_4| > |0.433.V_6|$$

2.6 Servo performance

The overall system, modelled in CSMP, combines equations (3) to (10). Figure 7 shows the response to the speed profile which, given F = 0.6 (equation (4)), corresponds with tests on the hardware.

The model also gives a predicted response for 2 g flying conditions. (See figure 8). Speed fluctuations above that demanded are due to gravity acting on the unbalanced masses and are evident in both responses. The effect is more pronounced in figure 8 as would be expected.

3. COMPUTATION OF FORCE AND CALCULATION OF SPEED

Total vertical force component is the sum of vertical components generated by each mass expressed as follows:-

$$M.R.\text{Cos}\theta.(\omega)^2$$

Mass angular velocity is ω and displacement is θ, referred to bottom dead centre. Positive force is downwards but inversion in the analogue amplifier gives positive volts for positive force upwards. The following expression refers each ω to ω_a, velocity of mass A.

$$M.R.N^2.\text{Cos}\theta.(\omega_a)^2 = K.\text{Cos}\theta.(\omega_a)^2$$

where N is the gear ratio and K a fixed multiplier.

Look-up tables "K.Cosθ", one per mass, are read for each θ given by the appropriate angle counter. These terms are summed and then multiplied twice by the common factor ω_a to give F_v vertical component of force (see equation (11)).

$$F_v = \omega_a^2[M_aR_a\text{Cos } \theta_a + M_bR_bN_{ba}^2\text{Cos } \theta_b + M_cR_cN_{ca}^2\text{Cos } \theta_c + M_dR_dN_{da}^2\text{Cos } \theta_d]$$

where $N_{ba} = 1.01$ gear ratio mass B:mass A

$N_{ca} = 2.88$ gear ratio mass C:mass A

$N_{da} = 2.89$ gear ratio mass D:mass A

$$F_v = \omega_a^2[0.3471 \text{ Cos } \theta_a + 0.4560 \text{ Cos } \theta_b + 0.3666 \text{ Cos } \theta_c + 0.4719 \text{ Cos } \theta_d] \quad (11)$$

3.1 Data processing

Each mass A, B, C and D generates a pulse at bottom dead centre (bdc) while the motor pick-off generates 20 pulses/rev (figure 2). Alternate motor pulses "E", together with the A, B, C, and D pulses, interrupt the program at precise instants providing all data to compute force and calculate speed. The motor pulse binary divider is reset by every D pulse to avoid division ambiguity when generating E pulses. These define increments "θ" and initiate computation of instantaneous force.

Angle counters (A, B, C and D), indexed every E pulse, are reset by the respective A, B, C or D pulse. Precisely 260 increments of 1.38°/rev of A and 90 increments of 4°/rev of D occur. A non-integer number of increments occur for B and C thus a fraction of an increment is carried over to the next revolution. The resultant error is more serious in C and is corrected by the following additional term.

$$0.3666.\delta.\text{Sin } \theta_c$$

where δ is the angle error for the given revolution of C. (RCNT)

A register initially zeroed is incremented by each A pulse. This is designated Motor Speed Pointer (MSP) and indexes tabled demand speed to give a speed profile conforming to equations (1) and (2)).

3.2 The microprocessor

The program, stored in a 2 kbyte EPROM, is processed by an INTEL 8085 CPU. Interrupts are controlled by an INTEL 8259 "Programmable Interrupt Controller" and then directed to the 8085 in priority order and with preset vectored addresses. Figure 6 is the system diagram.

Coincidence of A and B outputs a pulse from "AND" gate 11 to give interrupt CI and pressing "run" generates interrupt S. Together with A, B, C and D, the 8259 handles 7 interrupts. All, except S are finite width monostable controlled pulses. A and B, 2.7 ms wide, avoid any overlap outside true coincidence but detect CI with maximum pick-off misalignment. E, 100 μs wide, gives good separation at maximum speed.

C and D, 5 ms wide, are not critical. CI is the overlap of A and B. Highest priority is given to CI to cause an immediate stop. Lowest priority for E ensures that any angle counter is reset before force is computed so removing a possible phase error.

Mass A angular velocity is measured by timing the E pulse period using an INTEL 8253 "Programmable Interval Timer". The sum of cosines is then divided by the timer contents using an INTEL 8231 16-bit "Arithmetic Processor". An INTEL 8155 provides: the 20 RAM locations for flags and temporary stores; two 8-bit output Ports for velocity and force respectively; one 6-bit output Port to control lamps and relays and a divider for clock frequency conversion to 20 kHz. Analogue force and velocity demand are output from two AD 558 8-bit digital to analogue converters.

Power to the microprocessor ±12 V and +5 V is taken from local series regulators connected to the additional aircraft dc supplies, ±20 V and +9 V. Buffers convert the TTL outputs to 28 V for lamps and relays. Force waveform smoothing uses an operational amplifier giving an output in the range -5 V to +5 V.

3.3 Sequence of operation

Power on initialises the system, setting peripheral operating conditions and running the motor at low speed. Power-on, initialisation and motor running lamps glow and "run" demand is inhibited. Interrupts are serviced to compute force and detect overspeed. Loss of interrupt pulses or overspeed raises ABORT to cut off power to the system.

Coincidence CI stops the motor, power-on and motor brake lamps glow and the "run" demand is allowed.

The "run" demand S starts the motor. Interrupts except E and CI are serviced then, after one revolution of mass A, interrupts E will be serviced. This allows the counters to acquire a correct measure of speed before force is computed. After 8 rev of mass A interrupt CI will be serviced. The reason for inhibiting CI initially is to prevent a false stop immediately after start which could occur if masses A and B had rocked back to a position before coincidence prior to start. Motor speed demand is a function of the A interrupt count MSP. Loss of A, B or E pulses or overspeed raises ABORT, cutting off system power. During "run" power-on and motor running lamps glow.

ABORT cuts off power and brakes the motor causing the Abort lamp to glow. The system must be reinitialised by switching power off and on.

3.4 Software

The software contains 7 interrupt service routines, each being executed without interrupt from the other 6. A called routine INIT is serviced once following power-on. On completion of each routine the program returns to the EXECUTIVE. Look-up tables for speed, sine and cosine are provided. Compiled in 8085 assembler language, the program runs fast enough to achieve the required computing speed.

The software structure is shown in figure 9 and a flow chart of EXECUTIVE in figure 10. Figure 17 shows the timing of interrupts A, B, C and D at run start and figure 18 shows E pulse interrupt and execution timing during the highest speed part of the run.

3.4.1 Initialisation

At power-on, the program counter starts the instruction sequence from zero, setting the stack pointer and entering EXEC. EXEC calls INIT and, on return, waits for interrupts. INIT sets those conditions shown in figure 10.

The 8259 is programmed with vector address list 1; A is serviced by ISRA1 fixing mass A speed at 2 rev/s. Operating modes of the three timers, designated first, second and third in the 8253 (figure 6), are set to a limiting count of 256 and to allow gate reset. Angle counters are zeroed, MSP and RCNT are initialised. Other internal counters are initialised. An overspeed limit of 4 rev/s applies during initialisation.

The three ports in the 8155 (figure 6) are set respectively;

(1) 8-bit output for Computed Force,

(2) 8-bit output for Demand Speed, and

(3) 6-bit output for lamps, motor relay, abort and 1st/2nd timer switch.

Force output and Demand Speed are set to zero volts, initialise and run lamps are illuminated and the motor relay is operated. A bias in the velocity control (see figure 5 pin 9) during initialisation raises the demand speed to 2 rev/s.

The program returns to EXEC and halts to wait for interrupts. Force is computed while the system runs at slow speed until ended by CI.

3.4.2 Coincidence ISRCI

Coincidence of A and B pulses generates interrupt CI (see figure 12) defining the end of initialisation or run and the start point for the next run.

ISRCI resets angle counters and all other internal counters. MSP and RCNT are zeroed. The motor is dynamically braked, power-on and motor brake lamps are illuminated. Interrupts except S are masked and the stack pointer is set to the top of the stack.

3.4.3 Run ISRS

Pressing "run" generates interrupt S (see figure 13). The 8259 is now programmed with vector list 2 for A to be serviced by ISRA to vary the demand speed. Initial demand speed is set to 2.5 rev/s and the over-speed limit 8 rev/s both referred to mass A.

The E interrupt occurrence counter ISREOC is zeroed. All interrupts except E, S, and CI are unmasked (ref.4.3).

Program returns to the executive and halts to wait for interrupts. As the motor rotates force is computed and demand speed varied until ISRCI ends the run.

3.4.4 Mass A at bottom dead centre ISRA and ISRA1

Interrupt A calls ISRA during run (figure 14(a), list 2) and ISRA1 during initialisation (figure 14(b), list 1). Angle counter A is zeroed and interrupt A is masked by both routines. Masking, removed when angle counter A reaches 355°, gives protection against false A pulses. The angle counter is incremented in ISRE.

Interrupt E, masked by ISRS, is unmasked after the first revolution of mass A in ISRA. Since no masking occurred in INIT, unmasking in ISRA1 is unnecessary.

The number of revolutions made by mass C beyond CI before run start has to be computed. This number is entered into RCNT to correct phase errors (ref.4.1). Computation is made on the second occurrence of ISRA when the mass B angle counter gives a measure of angular difference between masses A and B. The value for RCNT is given in Table 2.

TABLE 2. PHASE CORRECTION

A rev wrt Coincidence	B Angle Count	RCNT
-1 to 0	1	1
0 to 1	3	3
1 to 2	6	6
2 to 3	8	9
3 to 4	11	12

ISRA increments MSP and fetches demand speed from the look-up table SPEED. Demand speed is output via port (2). ISRA1 also increments MSP to enable detection of loss of E pulses during initialisation.

Interrupt CI, masked by ISRS, is unmasked on 8th revolution of mass A. Since no masking occurred in INIT, unmasking in ISRA1 is unnecessary.

3.4.5 Mass B at bottom dead centre ISRB

Interrupt B calls ISRB (figure 15). Mass B angle counter is zeroed and B interrupts are masked. Unmasking occurs when the angle counter, incremented by ISRE, reaches 355°.

3.4.6 Mass C at bottom dead centre ISRC

Interrupt C calls ISRC (figure 15). Mass C angle counter is zeroed and C interrupts are masked. Unmasking occurs when the angle counter, incremented by ISRE, reaches 356°.

RCNT is incremented to act as a pointer to the phase error (δ) correction look-up table (ref.4.1). This error cycles every 19 rev of C so RCNT is reset to 1 on a count of 19.

3.4.7 Mass D at bottom dead centre ISRD

Interrupt D calls ISRD (figure 15). Mass D angle counter is zeroed and D interrupts are masked. Unmasking occurs when the angle counter, incremented by ISRE, reaches 356°.

3.4.8 Force computation ISRE

Pulse E requests ISRE (figure 16). ISRE contains 4 subroutines: TIME, COSGEN, DIVIDE and ENDE. These are run in sequence although TIME is fully executed only once every 6th ISRE.

(a) Subroutine TIME

The duration of 6 successive E intervals is totalised in either of two timer/counters C_1 clocked at 20 kHz. This count defines ω_a as follows:-

$$C_1\omega_a = 2900 \qquad (12)$$

The first and second timers alternately store or count over the period. Excess speed, indicated by a low count, is flagged and 10 sequential flags raise Abort.

TIME increments the occurrence counter ISREOC which is reset to one every 6th ISRE. On reset, the read and count functions of the first and second timers are reversed, the count for the previous 6 E-pulse period being stored at location TIMOUT for later recall in DIVIDE. For the other 5 ISRE occurrences, the program jumps to COSGEN using the same TIMOUT value.

After loading TIMOUT, program jumps to COSGEN unless the count is below the limit set in INIT or ISRS. For a low count (overspeed), a test is made for sequential occurrence. This test divides the mass D angle count by 6 (6 E-pulse periods) and compares with the contents of SSTIC (Sequence of Short Time Intervals Counter). For overspeed SSTIC is incremented every 6th E pulse and the two counts remain equal. Each time equality is detected STIC (Short Time Interval Counter) is incremented. If the two counts are unequal SSTIC is loaded with mass D angle count divided by 6 and STIC zeroed. If the counts are equal on 10 successive occasions, STIC count reaches 10 and ABORT is raised.

Mass D angle counter is zeroed at count of 90, so if overspeed were detected at a count of more then 30 it would again be zeroed before a sequence of 10 could be counted. However, if overspeed continues, it will be detected during the next revolution of mass D. If low count occurs the program jumps to ENDE and the previous TIMOUT is used for the next 5 ISRE executions.

(b) Subroutine COSGEN

Equation (11) mass.radius.cosine values are read from indexed look-up tables. When integer coefficients (Table 3) are assigned to these terms, in place of those in equation (11), the sum of terms (SUM), dimensioned: kg.m, is defined by the digital count: C_2 in equation (13).

TABLE 3. COSINE AND SINE COEFFICIENTS

Name	Coefficient	Integer	Remarks
STADA	0.3471	Ka = 92	Mass A Cosine
STADB	0.4560	Kb = 122	Mass B Cosine
STADCC	0.3666	Kc = 99	Mass C Cosine
STADD	0.4719	Kd = 127	Mass D Cosine
STADS	δ*0.3666	Kcc = 6	Mass C Sine Correction

$$C_2 = SUM.263 \qquad (13)$$

Integers in Table 3 make maximum use of data space, one byte per integer coefficient. SUM uses 2 bytes. The sine correction integer coefficient K(cc) is multiplied by 1, 0.5 or 0 using right shifts. This approximates δ (Section 4.1) and the number of shifts, function of RCNT, is selected from table STADCT.

(c) Subroutine DIVIDE

The Arithmetic Processing Unit (APU) divides 16-bit integers:

$$\frac{K.C_2}{C_1.C_1} = \frac{K.263}{2900.2900}.SUM.\omega_a^2 \quad \text{bits/N} \qquad (14)$$

Making K = 4096 moves the numerator up into the 10 most significant bits of the 16-bit register, optimising division accuracy. The APU status, continually tested during and after the 30 µs division time, indicates excess time. This is treated as an APU failure and gives a release. Divide performs the following sequential functions.

(1) If division counter < 12 use default value of time.

(2) Enter C_2 into APU (Dividend).

(3) Enter C_1, measured or default value, into APU (Divisor).

(4) Output divide command to APU

(5) Test for Division complete and read Answer.

(6) Rescale Answer to fill most significant of 16 bits and reenter as dividend.

(7) Reenter C_1 as Divisor.

(8) Output divide command to APU

(9) Test for division complete and read Answer.

(10) Scale and convert offset binary to binary.

(11) Output force to digital to analogue converter.

D to A conversion, 25.6 bits/V, gives a scale factor defined as follows:

$$\frac{2900 \times 2900 \times 25.6}{263 \times 4096} = 199 \quad \text{N/V}$$

(d) Subroutine ENDE

ENDE increments angle counters A, B, C and D. When angle counts for masses A and B exceed 355° the respective interrupt is unmasked. When angle counts for masses C and D exceed 356° the respective interrupt is unmasked. If any count exceeds 360° it is restored to 360° and any overcount for masses A or B is recorded; more than 3 overcounts raising an Abort. For counts to exceed 360° it indicates loss of A, B, C or D pulses.

3.4.9 Loss of E pulses

Loss of E pulses indicates a possible control system failure which could cause the motor to run at high speed. It is a hazardous condition requiring an automatic abort.

The third timer (8253) is clocked at 20 kHz and reset by each E pulse at a maximum rate of 520/s. This gives a maximum count of 38 before reset. With E pulse loss, an eventual count of 256 interrupts the CPU to abort the system.

3.4.10 Overspeed

Any other control failure could cause the motor to run at high speed and protection is needed. Minimum E pulse periods are set by program as a limit on measured speed. This limit must be exceeded for 60 consecutive E pulses for overspeed to be confirmed.

Limit speeds referred to mass A are: 4 rev/s (initialisation) and 8 rev/s (run). A proposed modification sets a profiled overspeed limit of 20% at low speed reducing to 12% at high speed.

3.4.11 Loss of A or B pulses

The loss of either A or B pulses prevents coincidence detection allowing the motor to run indefinitely. Failure of A pulses stops MSP indexing and fixes demand speed; the worst case being a failure of A at high speed. Loss of either pulse is hazardous.

During ENDE, an overcount (>360°) of angle A or B indicates loss of A or B pulse and is recorded. More than 3 occurrences increments MSP and raises Abort.

3.4.12 Abort

Software aborts, rejected for MSP counts less than 2, occur for overspeed or loss of A, B or E pulses. Abort masks all interrupts and outputs a signal to operate the abort relay. Program halts, the relay removes power from the store and the motor is dynamically braked. Interrupt masking prevents any possible removal of abort by a service routine before the relay operates.

3.4.13 Overrun

Failure to stop at coincidence when MSP is zeroed means that, during the next run, MSP does not define accurately the mass A revolution count. Thus the demand speed profile slips with respect to the disposition of masses causing an increased maximum force during acceleration and a reduced maximum force during deceleration. Limitation on force output occurs for a displacement of more than 4 turns of mass A.

Environmental tests showed overrun never to exceed 1 turn of mass A and is considered unlikely ever to do so. A modification to compensate for overrun is in process of development should it ever be required.

3.4.14 System outputs

Port 1: Force, output during initialisation and run, has a limited range of 1000 N for steel masses.

Port 2: Demand speed, constant 2 rev/s during initialisation and variable 2.5 to 7 rev/s during run (Speeds referred to mass A).

Port 3: Motor run relay, Abort relay and Indicator lamps. Power fail releases Abort relay setting Abort.

4. MANUAL CONTROLS

Controls, located in the cockpit, are Power-on, Run and Abort. The following coloured indicator lamps are visible to the pilot: Power-on (yellow), Run (green), Initialise (clear), Brake (blue) and Abort (red).

Power-on connects the 400 Hz, +28 V dc, ±20 V dc and +9 V dc supplies to the store. Initialisation is indicated by Run and Init lamps both on. On completion these lamps are off and Brake is on. When Power-on and Brake lamps are both on the system is ready to run.

Run command is confirmed by Run lamp on and Brake lamp off. The duration of a run should not exceed 28 s otherwise a fault exists.

Abort command, confirmed by Abort lamp, removes all power except +28 V dc. This command, either manually or software originated, is self locking and can be released only by Power-off followed by Power-on. The system is then reinitialised.

A possible modification retains microprocessor power after Abort to enable Abort conditions to be telemetered to ground. This requires significant changes and, as only three conditions give software Abort, was not considered necessary.

5. ERRORS IN COMPUTING FORCE

5.1 Phase jitter

The numerical value of force is computed for every E pulse. After outputting force angular displacements θ are incremented for each mass. Computing time between the E pulse and force output varies between 460 and 500 µs. This gives a phase jitter of 0.7° at the highest frequency and is trivial by comparison with other errors.

5.2 Cosine terms quantisation error

Mass.radius.cosine values in the look-up tables are 2's complement 7 bit integers. All four tables have a common scale factor 0.00371 per least significant bit (lsb). The highest coefficient in equation (11), 0.4719 for mass D, sets scaling for optimum space usage. Maximum quantisation error (one half lsb) is thus ±0.00185 for each value.

SUM has a maximum value at start of 1.64 scaled to decimal integer 442. An estimate of probable error after summation is 0.0026 or ±0.15% of start value. In the middle of the run masses are in antiphase and maximum SUM reduces to 0.16. Probable quantisation error at this point is ±1.5% of peak force generated. Maximum peak force occurs at the one third point and maximum instantaneous quantisation error is less than 1% of it.

Such errors are random and the time integral of error tends to zero. They are reduced by smoothing whose effect is greatest in the middle of the run where the E pulse rate is highest and the error is largest.

5.3 Cosine terms angular error

Initially centres of gravity of all masses are at bdc from which angular displacements are counted by E pulses; 260 for each A and 90 for each D revolution. Gearing to B and C gives non-integer counts for each revolution. Therefore on resetting counters B and C at 360° subsequent displacements have slip errors: up to 1.38° for B and up to 4° for C.

Masses A and B act as a pair, the resultant force being the vector sum. If mass B vector has a phase error, the sum vector has a gain and phase error, maximum when A and B are in antiphase. Phase error examples are 1.38° error in B giving maximum 2.8° error in the A+B sum vector and 4.0° error in C giving maximum 12.0° error in the C+D sum vector. These errors occur at mid run. Means are provided to reduce the C+D gain and phase errors; those for A+B being tolerable.

Given an error δ in measured displacement of mass C:

$$\cos(\theta_c + \delta) = \cos\theta_c - \delta \sin\theta_c$$

Provision of a Sine table weighted with three possible values of δ and added to SUM gives adequate correction. Values of δ are a function of revolutions of mass C from start (RCNT). RCNT, initially computed in the second occurrence of ISRA, is indexed each C pulse.

Maximum phase errors are 2.8° at 7 Hz and 2.5° at 20 Hz.

5.4 Angle definition errors

Angular misalignment of four centres of gravity, masses A, B, C and D, at bottom dead centre and the four pickoff's does not exceed 0.5°. No alignment is provided for the E pulse generator which measures displacements from bdc.

The tacho-generator outputs 20 pulses/motor revolution and, divided by 2, provides E pulses. The divider is set, once per revolution of mass D, to remove division ambiguity. Therefore maximum angle error of E pulses with respect to mass D is -2° and to mass A is -0.68° but probable errors are half these figures.

Force, computed for each E pulse, is converted to a staircase waveform whose average instantaneous value lags by one half an E pulse period. This gives a 0.68° lag in the low band, 2.5 to 7.0 Hz, and a 2.0° lag in the high band, 7.0 to 20.0 Hz. Look-up tables are adjusted effectively to advance all angles by one E pulse period converting all lags to leads. Thus, when the misalignment errors are added to staircase errors the maximum total becomes +2° for the high band and +0.68° for the low band. The probable errors are now +1.0° and +0.34° respectively.

5.5 Division errors

SUM is a 10-bit integer raised to 16 bits for the first dividend. The divisor, TIMOUT, is an 8-bit integer from the E pulse period count. Division yields a 9-bit integer raised to 16 bits for the second dividend.

TIMOUT, used again for the next division, yields a 9-bit integer reduced to 8 bits for D to A conversion.

Rounding errors after division are negative but rounding errors in TIMOUT, also negative, give positive errors after division. Since two divisions are made, divisor errors are more serious tending to more than compensate divider rounding errors.

Table 4 gives worst case conditions, zero dividend error and maximum divisor error, at 3 points on the run; low, mid and high speed.

TABLE 4. DIVISION ERRORS

ω Rad/s	E Freq	6 Periods	Count	Rounded Count	Error	Final Error
15.7	650 Hz	9.249 ms	184.98	184	0.545%	1.09%
32.8	1360 Hz	4.499 ms	89.98	89	1.09%	2.18%
43.3	1790 Hz	3.349 ms	66.98	66	1.52%	3.04%

Maximum force 746 N occurs at 32.8 rad/s and, given an error of 2.18% due to divisor rounding, becomes 762 N. Further rounding in the final division by two tends to reduce the error. Maximum rounding errors occur with minimum probability and errors in successive counts integrate down to 0.5%. Analogue smoothing reduces the error due to divisor rounding to less than 14 N anywhere in the trace.

At points on the trace away from the peaks, smaller dividends have larger rounding errors tending to overcompensate divisor errors.

5.6 Estimate of total errors

A maximum quantisation error of 1.5% was predicted at the high frequency part of the run. This is a percentage of the maximum peak force, 400 N, in the near vicinity and amounted to 6 N. Allowing for analogue smoothing a reduction to 4 N can be expected. Rounding error (3.04% of 400 N) amounts to 12 N giving a total of 16 N at high frequency.

At mid-frequency, quantisation error (0.6% of 750 N) is 4.5 N reduced to 4 N by analogue smoothing. With rounding errors added, (2.18% of 750 N) the predicted total error is 20 N.

At low frequency, quantisation error (0.15% of 400 N) is 0.6 N and a rounding error (1.09% of 400 N) gives a total of 5 N.

6. FUNCTIONAL TESTING

6.1 Test unit

The microprocessor normally receives interrupt pulses from the mechanical module. A test unit simulates the module by generating pulses at the same rates and in the same sequence. Pulse rates may be increased to simulate overspeed and stopped to simulate failures. The unit is initialised and run as though it were the module. Although it sets its own speed profile, not responding to any speed demand, it is adequate to test all the functions of the microprocessor.

The cable from the module carrying pulses A, B, C, D and E as well as controls S, INIT, RUN and ABORT is disconnected and replaced by a cable to the test unit. For "in situ" tests, the pilot's controller is used and, for laboratory tests, a simulated controller is used.

The test unit contains its own microprocessor which self initialises with power on. It reacts to signals at the parallel input port, INIT, RUN or ABORT by generating interrupt pulses. Push buttons on the test unit increase pulse rates or stop any of the pulses. The pulse sequence starts and stops when all pulses coincide.

Pulses are derived from a common clock feeding into 5 dividers. The dividers generate pulses of width equal to equivalent pulses from the mechanical module. Speed is varied by changing the common clock frequency.

After CI, 99 A, 100 B, 285 C, 286 D and 25740 E pulses are generated before the next CI. The lowest common multiple being 2 445 300, dividers for A, B, C, D and E are set respectively to 24, 700, 24 453 8580, 8550 and 95. During initialisation, the clock frequency is 40 000 pulses/s but during run it varies between 60 000 and 175 000 pulses/s approximately. To test overspeed the clock frequency is raised to 200 000 pulses/s.

A numerical display of wheel A revolutions from start appears on the tester front panel.

6.2 Pulse timing

Figure 17, timing diagram, shows interrupt pulses A, B, C and D during the first part of a run. All coincide at start, then B leads A and D leads C by increasing intervals until the end when all coincide again. During a run B and D coincide 5 times.

Figure 18 shows ISRE timing during the highest speed section of a run. Average execution time is 460 μs allowing an average 90 μs gap before the next E interrupt in which to service A, B, C or D. The longest, ISRA, executes in 75 μs and, when B and D coincide, ISRB and ISRD execute sequentially in 50 μs. The gap after every 6th ISRE is 45 μs and interrupts occurring here dalay executing the next ISRE causing a time jitter inin computed force. During one revolution of mass A, 260 E interrupts are serviced outputting 260 points on a force time series. Only 8 additional interrupts occur in this period and the resultant jitter is insignificant.

6.3 Force waveform

The first half of the force waveform F2 is given in figure 19. It was generated by the processor when connected to the test unit. Variations in waveshape occur if the run starts more than one revolution of wheel A from coincidence or if speed fluctuates.

To measure actual force generated, the module was set onto a test bed with calibrated strain gauges in the support brackets. A waveform similar to F1 was output by the gauges. A 100 Hz superimposed resonance due to the module mass and support stiffness, made impossible any comparison in the time domain. However fourier transforming the two force waveforms F1 and F2 enabled comparison in frequency domain between 2.5 and 20 Hz isolating the 100 Hz ripple.

6.4 Spectrum analysis

F1 and F2 are fed respectively to the two inputs of a dual channel spectrum analyser HP 3582. They are simultaneously sampled over 4 consecutive periods of 5 s. Each sampled time series is then transformed into 4 spectra and averaged into a single spectrum. This covers the first 20 s of a 28 s run.

To cover the whole run, F1 and F2 are restarted and again sampled simultaueously over 4 periods of 5 s but delayed by 8 s. The resultant spectrum is then the average of 8 spectra.

The transfer function between the two force waveforms F1 and F2, indicates the error in terms of gain and phase of computed force over the useful spectrum 2.5 to 20 Hz. Instrument errors are removed by reversing the channels for F1 and F2. The two transfer functions, forward F1:F2 and reverse F2:F1, are shown in figure 20. This confirms the predicted errors of ±5% gain and ±5° phase and a group delay or transport lag of 850 μs.

7. DESIGN COMMENTS

The flutter controller and force computer were designed to meet the requirements of a mechanical system described in other memoranda. Force measurement by load cells was discounted since these respond also to aerodynamic reaction forces. A continuous output of computed force demanded minimum execution times and maximum rates of execution which were achieved by programming in assembler language.

The use of a series wound universal ac/dc motor was dictated by considerations of power and size. It provided a one sided control which worked satisfactorily in the module although not a perfect system.

Thyristor control required considerable filtering to prevent electrical interference with the digital electronics and with the aircraft systems.

After some development all optoelectronic devices worked well. The main problems were maintaining good reflective surfaces and avoiding induced electrical noise at the device inputs. This was particularly the case with the motor shaft pick-off in the presence of commutator noise.

Exhaustive tests were carried out to prove software design. Without the test unit many of the software errors would not have been revealed. Additional faults were revealed during vibration testing in both software and hardware. These were all corrected by modifications.

8. CONCLUSIONS

Data acquisition, at least 90 points/cycle, ensured the accuracy of the force waveform with a minimum of smoothing. Assembler language programming minimised execution times enabling this to be achieved.

The system has been installed in a Harpoon round under the wing of an F111 and has given satisfactory performance for over 200 sorties.

NOTATION

A,B,C,D	mass or interrupt designation
CI	coincidence interrupt
C_1, C_2	time count, sum of cosines scaled integers
E	compute force interrupt
F_v	vertical component of force (N)
F	coulomb friction (N)
K	scaling factor 4096
M_a, M_b, M_c, M_d	mass A, mass B, mass C, mass D (kg)
MSP	motor speed pointer
N	gear ratio
R_a, R_b, R_c, R_d	radii to c of g: mass A, mass B, mass C, mass D (m)
RCNT	mass C revolution counter
S	start interrupt signal
s	laplace operator
SUM	sum of integers (kg.m.cosine)
T	torque on motor shaft (Nm)
t	time (s)
V_1	servo amplifier output (V)
V_2	motor drive (V)
V_4	motor demand speed (V)
V_5	frequency to voltage converter output (V)
V_6	motor output speed (V)
δ	error angle mass C (rad)
θ_a, θ_b, θ_c, θ_d	displacements from bottom dead centre, masses A, B, C, D (rad)
θ_o	displacement of motor shaft (rad)
ω_o	velocity of motor shaft (rad/s)
α_o	acceleration of motor shaft (rad/s/s)
ω_a	velocity of mass A (rad/s)

APPENDIX I

PROGRAM LISTING

```
NAME    H1R3V6
; VERSION 3.6 OF FLUTTER EXCITER MODULE CONTROL SOFTWARE
;---------------------------------------------------------
; THE ASSEMBLED PROGRAM:  "H1R3V6" INCLUDES "SYMBOL.2V3" (SYMBOL DEFINITION) &
; "H2R3V6".  H1R3V6 CONTAINS THE EXECUTIVE, INITIALISATION PLUS 3 SUBROUTINES:
; COINCIDENCE, START & ISRE. H2R3V6 CONTAINS 6 SUBROUTINES: ISRA, ISRA1, ISRB,
; IRSC, ISRD & LEP2 AS WELL AS LOOK UP TABLES AND CALL LISTS.  ALL SUBROUTINES
; RESPOND TO INTERRUPTS & RETURN TO EXEC.
; POWER-ON STARTS INITIALISATION ROUTINE,  JUMPS TO EXEC, ENABLES INTERRUPTS &
; HALTS. INITIALISATION SETS A MOTOR SPEED OF 2 R.P.M., REFERRED TOMASS A., &
; RUNS MOTOR. WHILE MOTOR RUNS FORCE IS COMPUTED; STOPPING WHEN COINCIDENCE IS
; DETECTED.
; IN THE INITIAL PERIOD INTERRUPTS, EXCEPT START, ARE UNMASKED.   AN ABORT CAN
; OCCUR FOR OVERSPEED (4 R.P.M), LOSS OF A,B OR E PULSE OR C.P.U. FAILURE.  AT
; COINCIDENCE ALL INTERRUPTS, EXCEPT START, ARE MASKED & A RUN CAN BE STARTED.
; INTERRUPT SERVICE ROUTINES: ISRA1, ISRB, ISRC, ISRD, ISRE, & ISRCI CALLED BY
; THE PROGRAMMABLE INTERRUPT CONTROLLER.
; IN THE RUN PERIOD INTERRUPTS, EXCEPT START, ARE UNMASKED. AN ABORT CAN OCCUR
; FOR OVERSPEED (8 R.P.M), LOSS OF A,B OR E PULSES OR C.P.U. FAILURE.  A SPEED
; PROFILE & COMPUTED FORCE ARE OUTPUT. INTERRUPT SERVICE ROUTINES: ISRA, ISRB,
; ISRC, ISRD, ISRE & ISRCI CALLED BY THE P.I.C.
; DYNAMIC MASKING PROTECTS AGAINST RANDOM PULSES ON A, B, C & D LINES. SPECIAL
; HARDWARE DETECTS LOSS OF E PULSES; LOSS OF A & B PULSES BEING DETECTED IN THE
; SOFTWARE.
;
; --------------------MEMORY MAP--------------------
;
; 0000H TO 03A3H        R.O.M.   PROGRAM
; 03E0H TO 0785H        R.O.M.   TABLES
; 07C0H TO 07FCH        R.O.M.   CALL LISTS
; 0800H TO 0810H        R.A.M.   ACTIVE MEMORY
; 08C0H TO 08FFH        R.A.M.   STACK
; 0900H TO 0903H        8155     PORTS
; 1000H TO 1001H        8259     INTERRUPT CONTROLLER
; 1800H TO 1803H        8253     TIMER/COUNTER
; 2000H TO 2001H        8231     ARITHMETIC PROCESSOR
;
;--------------------------MODIFICATION HISTORY-------------------------
;
; --MOD.NO.1-- MASK ALL INTERRRUPTS AFTER ABORT R.W.L. 25-05-83
; --MOD.NO.2-- TEST INT.SERV.REG. (PIC) DURING ISRA1  R.W.L. 25-05-83
; --MOD.NO.3-- OVERRUN DETECT REMOVED (WOBBLE EFFECTS) R.W.L. 15-06-83
; --MOD.NO.4-- A,B,C & D MASKS REFER TO WHEEL COUNTS R.W.L.  18-06-83
; --MOD.NO.5-- OVERRUN DETECT REPLACED (ANTI WOBBLE PROG) R.W.L. 16-06-83
; --MOD.NO.6-- OVERSPEED DETECT DURING INITIALISATION R.W.L. 25-06-83
; --MOD.NO.7-- OVERRUN DETECT REMOVED (WOBBLE EFFECTS) R.W.L. 28-06-83
; --MOD.NO.8-- PROTECT AGAINST A.P.U. FAILURE R.W.L. 08-08-83
; --MOD.NO.9-- PROTECT AGAINST L.E.P. DETECT FAILURE R.W.L. 03-02-84
; --MOD.NO.10- PROTECT AGAINST A & B PULSE FAILURE R.W.L. 05-02-84
; --MOD.NO.11- PROTECT AGAINST C.P.U. FAILURE R.W.L. 15-02-84
;
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;

; COMPILED BY P.M.SYKES AND R.W.LEVINGE
; SYSTEM DESIGNERS: R.W.LEVINGE, P.M.SYKES & T.SHULTZ
```

```
;
;
; SYMBOL DEFINITIONS;
;==================
;
ICW3    EQU     00D6H   ; INITIALISATION CONTROL WORD 1 FOR P.I.C. (AFTER CI)
ICW2    EQU     0007H   ;        "          "      "  2  "    "
ICW1    EQU     00F6H   ;        "          "      "  1  "    "
TACW    EQU     0014H   ; TIMER "0" CONTROL WORD
TBCW    EQU     0054H   ;   "   "1"    "      "
TCCW    EQU     0094H   ;   "   "2"    "      "
INIT    EQU     00DCH   ; INITIALISATION WITH TIMER COUNTER A (PORT C)
RUN     EQU     00DEH   ; RUN WITH TIMER COUNTER A (PORT C)
READY   EQU     00FFH   ; READY OUTPUT WORD THROUGH PORT C
RCNT    EQU     0800H   ; WHEEL C REV COUNTER
WCAL    EQU     0801H   ; ANGLE COUNTER WHEEL A LOWER BYTE
WCAU    EQU     0802H   ;   "      "      "   " UPPER BYTE
WCBL    EQU     0803H   ; ANGLE COUNTER WHEEL B LOWER BYTE
WCBU    EQU     0804H   ;   "      "      "   " UPPER BYTE
WCC     EQU     0805H   ; ANGLE COUNTER WHEEL C
WCD     EQU     0806H   ; ANGLE COUNTER WHEEL D
MSP     EQU     0807H   ; MOTOR SPEED POINTER (WHEEL A REV. COUNTER)
TIMOUT  EQU     0808H   ; "E" PULSE TIME INTERVAL REGISTER
OSPED   EQU     0809H   ; SPEED LIMIT STORE
STIC    EQU     080AH   ; SHORT TIME INTERVAL COUNTER
SSTIC   EQU     080BH   ; SEQUENCE OF SHORT TIME INTERVALS COUNTER
ISREOC  EQU     080CH   ; "E" INTERRUPT OCCUR COUNTER
SWAB    EQU     080DH   ; COUNTER A/B SWITCH
PORTC   EQU     080EH   ; PORT C OUTPUT DATA REGISTER
DVCNT   EQU     080FH   ; DIVISION COUNTER
ABSEQ   EQU     0810H   ; A/B INTERRUPT SEQUENCE FLAG
LEPC    EQU     0811H   ; LOSS OF E PULSE COUNTER
TSTK    EQU     08FFH   ; TOP OF STACK ADDRESS
PCR     EQU     0900H   ; I/O PORT COMMAND REGISTER ADDRESS
PRTA    EQU     0901H   ; PORT A ADDRESS
PRTB    EQU     0902H   ; PORT B ADDRESS
PRTC    EQU     0903H   ; PORT C ADDRESS
PICA    EQU     1000H   ; PROGRAMMABLE INTERRUPT CONTROLLER ICW1 & OCW2 ADDRESS
PICB    EQU     1001H   ;      "            "          "     ICW2 & OCW1 ADDRESS
CNTA    EQU     1800H   ; TIMER 0 MAXIMUM COUNT ADDRESS
CNTB    EQU     1801H   ;   "   1    "      "      "
CNTC    EQU     1802H   ;   "   2    "      "      "
CWRA    EQU     1803H   ; PROGRAMMABLE INTERVAL TIMER CONTROL WORD REG. ADDRESS
APUD    EQU     2000H   ; ARITHMETIC PROCESSING UNIT DATA ADDRESS
APUC    EQU     2001H   ;      "          "         "   CONTROL ADDRESS
;____________________________________________________________________
;
;-___________________________________________________________________
;
; MAIN ROUTINE: EXEC
; CALLED ROUTINE: INITL
; INTERRUPT SERVICE ROUTINES: ISRS,ISRCI,ISRE,ISRC,ISRA,ISRB,ISRD,ISRA1,LEP
; INPUTS: TIMER, A.P.U.
; OUTPUTS: SPEED, FORCE, INDICATORS, RELAYS, TIMER SWITCH, A.P.U. DATA
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
        ORG     0000H
;
        LXI     SP,TSTK ;DEFINE TOP OF STACK.
        JMP     EXEC
;
```

```
        ORG     0038H
;
        RST     0
        RST     0
        RST     0
;
        ORG     2CH
;
LEP:    CALL    LEP2     ; "LOSS OF E PULSES" ABORT AFTER 2 REVS OF WHEEL A
        RET
;
;_____________________________________________________________________
;
; EXECUTIVE
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
        ORG     0050H
;
EXEC:   CALL    INITL    ;INITILISATION.
POS15:  EI               ; ENABLE PROCESSOR INTERRUPTS
POS14:  HLT              ; WAIT FOR INTERRUPTS
        LXI     H,PICA   ;FOR P.I.C. OCW2
        MVI     M,20H    ; SEND EOI
        JMP     POS15
;_____________________________________________________________________________
;
; --"INITIALISATION"--
; CONTROL WORDS TO PERIPHERALS: P.I.T., PORTS & P.I.C.(TO LIST 1). OUTPUT FORCE
; ZERO, OUTPUT SPEED 2 R.P.M., INIT & RUN LAMPS ON, MOTOR RUN.  TIMER CLOCK SET
; TO 20 KHz & TIMER "A" TO COUNT.   WHEEL ANGLE COUNTERS: A, B, C & D SET TO 1.
; MOTOR SPEED POINTER TO 1.  SHORT TIME INTERVAL AND SEQUENCE COUNTERS TO ZERO.
; "E" OCCUR COUNTER ZERO. OVERSPEED LIMIT 4.0 R.P.M. "START" MASKED.
; INPUTS: NONE
; OUTPUTS: FORCE, SPEED, LAMPS & MOTOR RELAY, PERIPHERAL COMMANDS.
; INTERRUPT MASKS: S
; DESTROYS: ALL REGS.
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
INITL:  LXI     H,LEPC   ;FOR L.E.P. COUNTER
        MVI     M,00H    ; SET TO ZERO
        LXI     H,PICA   ;FOR P.I.C. (ICW1.)
        MVI     M,ICW3   ; SET 11010110 ≡L.ADD.(C0)≡EDGE.TR.≡ADI=4≡SING≡NO ICW4≡
        INX     H        ;FOR P.I.C. (ICW2 & OCW1.)
        MVI     M,ICW2   ; SET 00000111 ≡H.ADD.(07)≡
        MVI     M,82H    ; SET 10000010 ≡MASK (S)≡
        LXI     H,PCR    ;FOR I/O PORT COMMAND REG.
        MVI     M,0CFH   ; SET PORTS: A,B,C FOR O/P.
POS12:  LXI     H,PRTA   ;FOR PORT A (FORCE)
        MVI     M,80H    ; SET ZERO FORCE
        INX     H        ;FOR PORT B (SPEED)
        MVI     M,00H    ; SET INITIAL MOTOR SPEED
        INX     H        ;FOR PORT C
        MVI     M,INIT   ; SET 011100 ≡P.I.T. GO LOW≡ABORT OFF≡INIT ON≡ RUN MOTOR≡
        INX     H        ;FOR TIMER SECTION (L.S. BYTE OF COUNT)
        MVI     M,9AH    ; SET C.P.U. CLK. DIVIDER TO 154 FOR P.I.T. CLK.
        INX     H        ;FOR TIMER SECTION (H.S. BYTE OF COUNT & MODE)
        MVI     M,40H    ; SET FOR CONTINUOUS SQ. WAVE
        LXI     H,PCR    ;FOR TIMER SECTION (CONTROL)
        MVI     M,0CFH   ; START DIVIDER
        LXI     H,CWRA   ;FOR P.I.T. COUNTER CONTROL WORDS.
        MVI     M,TACW   ; SET. 00010100 ≡CNTR 0≡L.S.BYTE≡DIV. BY (N)≡PULSE ≡
```

```
        MVI     M,TBCW  ; SET. 01010100 ≡CNTR 1≡L.S.BYTE≡DIV. BY (N)≡PULSE ≡
        MVI     M,TCCW  ; SET. 10010100 ≡CNTR 2≡L.S.BYTE≡DIV. BY (N)≡PULSE ≡
        LXI     H,CNTA  ;FOR CNTR 0 (LOAD N)
        MVI     M,0FFH  ; SET N = FFH
        INX     H       ;FOR CNTR 1 (LOAD N)
        MVI     M,0FFH  ; SET N = FFH
        INX     H       ;FOR CNTR 2 (LOAD N)
        MVI     M,0FFH  ; SET N = FFH (12.8 msecs LOSS OF "E" PULSES)
        LXI     H,PORTC ;FOR PORTC DATA REGISTER
        MVI     M,INIT  ; STORE DATA
        LXI     H,0001H ;SET 1
        SHLD    WCAL    ; IN ANGLE COUNTER WHEEL A
        SHLD    WCBL    ; IN ANGLE COUNTER WHEEL B
        LXI     H,WCC   ;FOR ANGLE COUNTER WHEEL C
        MVI     M,01H   ; SET 1
        INX     H       ;FOR ANGLE COUNTER WHEEL D
        MVI     M,01H   ; SET 1
        LXI     H,RCNT  ;FOR REV COUNT WHEEL C
        MVI     M,01H   ; SET 1
        LXI     H,STIC  ;FOR SHORT TIME INTERVAL COUNTER
        MVI     M,00H   ; SET TO 0
        LXI     H,SSTIC ;FOR SEQUENCE OF SHORT TIME INTERVALS COUNTER
        MVI     M,00H
        LXI     H,SWAB  ;FOR COUNTER SWITCH
        MVI     M,055H  ; SET 01010101
        LXI     H,DVCNT ;FOR COMPUTATION COUNTER
        MVI     M,01H   ; SET INITIAL VALUE TO 1
        LXI     H,MSP   ;FOR MOTOR SPEED POINTER
        MVI     M,01H   ; SET TO 1
        LXI     H,ISREOC;FOR ISRE OCCUR COUNTER
        MVI     M,00H   ; SET TO 0
        LXI     H,OSPED ;FOR SPEED LIMIT STORE
        MVI     M,8CH   ; TIMOUT < 73H OR  4.0 REVS PER SECOND
        MVI     A,1EH   ; UNMASK RST5.5 TO ALLOW LOSS OF E PULSE DETECT
        SIM
;
        RET             ; RETURN TO EXEC
;______________________________________________________________________
; --INT. SERVICE ROUTINE 'PULSE COINCIDENCE'.--
; STOP MOTOR, RUN & INIT LAMPS OUT, FORCE OUTPUT TO ZERO, ANGLE COUNTERS,
; FLAGS & REGISTERS RESET.
; OUTPUTS: LAMPS, MOTOR, FORCE
; INTERRUPT MASKS: ALL EXCEPT S
; STACK POINTER TO TOP OF STACK
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
ISRCI:  MVI     A,1FH   ; MASK RST5.5 TO PREVENT LOSS OF E PULSE DETECT
        SIM
        LXI     H,PRTA  ;FOR PORT A (FORCE)
        MVI     M,80H   ; SET ZERO FORCE
        LXI     H,PRTC  ;FOR PORT C
        MVI     M,READY ; SET 111111 ≡P.I.T. GO HIGH≡ABORT OFF≡INIT OFF≡
        LXI     H,RCNT  ;FOR REV COUNTER WHEEL C.                 STOP MOTOR≡
        MVI     M,01H   ; INITIALISE TO 1.
        LXI     H,0001H ;   SET 1
        SHLD    WCAL    ; IN ANGLE COUNTER WHEEL A.
        SHLD    WCBL    ; IN ANGLE COUNTER WHEEL B.
        LXI     H,WCC   ;FOR ANGLE COUNTER WHEEL C.
        MVI     M,01H   ; SET TO 1
        INX     H       ;FOR ANGLE COUNTER WHEEL D.
        MVI     M,01H   ; SET TO 1
```

```
        LXI     H,MSP   ;FOR MOTOR SPEED POINTER.
        MVI     M,01H   ; SET TO 1
        LXI     H,PICA  ;FOR P.I.C. (ICW1)
        MVI     M,ICW3  ; SET 11010110 ≡L.ADD.(C0)≡EDGE TR.≡ADI=4≡SING≡NO ICW4≡
        INX     H       ;FOR P.I.C. (ICW2 &OCW1)
        MVI     M,ICW2  ; SET 00000111 ≡H.ADD.(07)≡
        MVI     M,0EDH  ; SET 11111101 ≡MASK E,B,A1,C,D & CI≡
        LXI     H,STIC  ;FOR SHORT TIME INTERVAL COUNTER
        MVI     M,00H   ; SET TO ZERO
        LXI     H,SSTIC ;FOR SEQUENCE OF SHORT TIME INTERVALS COUNTER
        MVI     M.00H   ; SET TO ZERO
        LXI     H,ABSEQ ;FOR A/B SEQUENCE FLAG
        MVI     M,55H   ; RESET FLAG OFF
        LXI     H,OSPED ;FOR OVERSPEED LIMIT STORE
        MVI     M,00H   ; NO LIMIT
;
        LXI     SP,TSTK ;FOR TOP OF STACK
        JMP     POS15   ; JUMP TO EXEC
;_________________________________________________________________________
;
; --INT. SERVICE ROUTINE 'START'.--
; SET P.I.C. TO LIST 2: 07E0H. RUN MOTOR AT BASE SPEED. SET OVERSPEED LIMIT
; TO 7.95 R.P.M.  RESET INITIALISATION FLAG. MASK INTERRUPTS: S & CI. RESET
; ISRE OCCUR COUNTER & MOTOR SPEED POINTER. RUN LAMP ON, ABORT & INIT LAMPS
; OFF.
; INPUTS: NONE
; OUTPUTS: MOTOR & LAMPS
; INTERRUPT MASKS: "S" MASKED
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
ISRS:   PUSH    PSW
        PUSH    H
        PUSH    D
        PUSH    B
;
        LXI     H,PICA  ;FOR P.I.C. (ICW1)
        MVI     M,ICW1  ; SET 11110110 ≡L.ADD.(E0)≡EDGE TR.≡ADI=4≡SING≡
        INX     H       ;FOR P.I.C. (ICW2 & OCW1)               NO ICW4≡
        MVI     M,ICW2  ; SET 00000111 ≡H.ADD.(07)≡
        MVI     M,83H   ; SET 10000011 ≡MASK(S,CI)≡
        LXI     H,ISREOC;FOR ISRE OCCUR COUNTER
        MVI     M,00H   ; SET TO 0
        LXI     H,PRTB  ;FOR PORT B (SPEED)
        MVI     M,00H   ; SET TO BASE SPEED.
        INX     H       ;FOR PORT C
        MVI     M,RUN   ; SET 011110 ≡P.I.T. GO LOW≡ABORT OFF≡INIT OFF≡
        LXI     H,PORTC ;FOR PORT C DATA REGISTER                MOTOR ON≡
        MVI     M,RUN   ; STORE DATA
        LXI     H,MSP   ;FOR MOTOR SPEED POINTER
        MVI     M,01H   ; SET TO 1
        LXI     H,OSPED ;FOR OVERSPEED LIMIT STORE
        MVI     M,0C5H  ; TIMOUT < 3AH OR 7.95 REVS PER SECOND
        MVI     A,1EH   ; UNMASK RST5.5 (ALLOW L.E.P. DETECT)
        SIM
;
        POP     B
        POP     D
        POP     H
        POP     PSW
        RET             ; RETURN TO EXEC
```

```
;___________________________________________________________________
;
; --INT. SERVICE ROUTINE 'E PULSES'.--
; CONTAINS 4 MODULES:  TIME, COSGEN, DIVIDE & ENDE.  TIME DEFINES ANGULAR
; VELOCITY, COSGEN RESOLVES FOUR M.R. PRODUCTS INTO VERTICAL AXIS, DIVIDE
; GIVES FINAL RESULT IN NEWTONS FORCE & ENDE INCREMENTS A, B, C & D ANGLE
; COUNTERS & UNMASKS A, B, C & D AT 355 DEGREES ANGLE.
; INPUTS: TIME COUNT & A.P.U. DIVISION.
; OUTPUTS: 8-BIT VALUE OF FORCE, A.P.U. COMMANDS & DATA, TIMER SWITCH.
; INTERRUPT MASKS: "A", "B", "C" & "D" UNMASKED.
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
ISRE:
;
;-__________________________________________________________________
;
; MEASURE TIME INTERVAL OVER 6 E PULSES BY READING CONTENTS OF ONE OF TWO
; COUNTERS.   MINIMUM TIME INTERVAL HAS BEEN SET FOR OVERSPEED DETECTION.
; MEASURED TIME VALUE IS USED FOR THE FOLLOWING 6 E PULSES, AFTER WHICH A
; NEW MEASURE IS TAKEN FROM THE SECOND OF TWO COUNTERS.
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
TIME:   LXI     H,ISREOC;FOR ISRE OCCUR COUNTER
        MOV     A,M      ; READ
        CPI     06H      ; SIX "E" PERIODS?.
        JZ      MESUR    ; MEASURE TIME AFTER EVERY SIXTH PERIOD
        INR     M        ; INC. ISRE OCCUR COUNTER
        JMP     COSGEN   ; JMP TO COSGEN.
MESUR:  MVI     M,01H    ; SET ISRE OCCUR COUNTER = 1.
        LXI     H,SWAB   ;FOR COUNTER SWITCH
        MOV     A,M      ; READ
        RRC
        MOV     M,A      ; REVERSE SWITCH SETTING
        INX     H        ;FOR PORT C DATA REGISTER
        MOV     A,M      ; DATA IN ACC.
        JC      L2       ; JUMP IF CARRY
        ANI     0DFH     ; REMOVE BIT 5 (COUNTER A HELD)
        MOV     M,A      ; PORT A DATA CHANGED
        LXI     H,PRTC   ;FOR PORT C
        MOV     M,A      ; FREE COUNTER (B) HOLD COUNTER (A)
        LXI     H,CNTA   ;FOR COUNTER (A)
        MOV     A,M      ; READ
        JMP     L3
L2:     ORI     08H      ; REPLACE BIT 5 (COUNTER B HELD)
        MOV     M,A      ; PORT A DATA CHANGED
        LXI     H,PRTC   ;FOR PORT C
        ORI     20H      ; CHANGE PORT C DATA
        MOV     M,A      ; FREE COUNTER (A) HOLD COUNTER (B)
        LXI     H,CNTB   ;FOR COUNTER (B)
        MOV     A,M      ; READ
L3:     CMA              ; (FFH - REMAINING COUNT) = TIME.
        LXI     H,OSPED  ;FOR OVERSPEED LIMIT STORE
        MOV     D,M      ; READ
        LXI     H,TIMOUT;FOR TIME REGISTER
        MOV     B,M      ; SAVE OLD VALUE OF TIMOUT
        MOV     M,A      ; STORE NEW VALUE OF TIMOUT
        ADD     D        ; ADD 2'S COMPLEMENT OF OVERSPEED LIMIT
        JC      COSGEN   ; CONTINUE IF TIMOUT < OR = LIMIT
```

```
;-----------------------------------------------------------------------
;
; TEST FOR SEQUENCE OF 10 SHORT TIME INTERVALS & ABORT.   OTHERWISE STORE
; PREVIOUS TIME VALUE & SKIP FORCE MEASUREMENT.
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
        MOV     M,B      ; USE PREVIOUS VALUE OF TIMOUT
        LXI     H,WCD    ;FOR ANGLE COUNTER WHEEL D
        MOV     A,M      ; READ
        MVI     B,00H    ; B REG TO HOLD (NO. OF "E" PULSES)/6
CNT:    INR     B
        SUI     06H      ; TEST DIVISION BY 6 ENDED
        JP      CNT      ; CONTINUE IF NOT
        LXI     H,SSTIC  ;FOR SEQUENCE OF SHORT TIME INTERVALS COUNTER
        INR     M        ; INCREMENT COUNTER
        MOV     A,B      ; (NO. OF "E" PULSES)/6
        SUB     M        ; COMPARE WITH COUNTER
        JZ      AB       ; JUMP IF SHORT TIME INTERVAL IS SEQUENTIAL
        MOV     M,B      ; EQUATE COUNTER TO (NO. OF "E" PULSES)/6
        LXI     H,STIC   ;FOR SHORT TIME INTERVAL COUNTER
        MVI     M,00H    ; SET TO ZERO
        JMP     ENDE
AB:     LXI     H,STIC   ;FOR SHORT TIME INTERVAL COUNTER
        INR     M        ; INCREMENT
        MOV     A,M      ; READ
        SUI     0AH      ; COUNT OF 10dec
        JP      ABORT    ; JUMP TO ABORT IF YES
        JMP     ENDE
;-------------------------------------------------------------------
;
;  COMPUTE FACTOR TO RESOLVE FORCE INTO VERTICAL DIRECTION BY SUMMING
;  4 COSINE TERMS (4 MASSES) & 1 SINE TERM (PHASE ERROR CORRECTION)
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
COSGEN: LHLD    WCAL     ; READ 16 BIT VALUE IN ANGLE COUNTER WHEEL A.
        LXI     B,STADA  ; START ADD. FOR COS(A) TABLE
        DAD     B        ;FOR COS(A)
        MOV     A,M      ; READ K(A)*COS(A)
        MVI     B,00H
        MOV     C,A      ; SAVE K(A)*COS(A) IN B,C REG.PAIR
        CPI     80H      ; CHECK SIGN OF COS(A)
        JC      POS3     ; JUMP IF POS.
        MVI     B,0FFH   ; 2s COMP IN UPPER BYTE
;
POS3:   LHLD    WCBL     ; READ 16 BIT VALUE IN ANGLE COUNTER WHEEL B
        LXI     D,STADB  ; START ADD. FOR COS(B) TABLE
        DAD     D        ;FOR COS(B)
        MOV     A,M      ; READ K(B)*COS(B)
        MVI     H,00H
        MOV     L,A      ; SAVE K(B)*COS(B) IN H,L REG. PAIR
        CPI     80H      ; CHECK SIGN OF COS(B)
        JC      POS4     ; JUMP IF POS.
        MVI     H,0FFH   ; 2s COMP IN UPPER BYTE
;
POS4:   DAD     B        ; K(A)*COS(A)+K(B)*COS(B)
        MOV     B,H      ; SAVE IN B,C REG.PAIR
        MOV     C,L
;
;K(C)*COS(C) WILL INCLUDE CORRECTION FOR PHASE ERROR
;
```

```
        PUSH    B         ; STORE RUNNING SUM ON STACK
        LXI     H,WCC     ;FOR ANGLE COUNTER WHEEL C.
        MOV     E,M       ; READ ANGLE LOW BYTE
        MVI     D,00H     ; ZERO HIGH BYTE
        PUSH    D         ; STORE ON STACK
        LXI     H,STADCS;FOR START ADD. SIN(C)
        DAD     D         ; ADD ANGLE
        MOV     C,M       ; READ K(C)*0.056*SIN(C)
        LXI     H,RCNT    ;FOR REV. COUNT.
        MOV     E,M       ; READ REV. COUNT LOW BYTE
        MVI     D,00H     ; ZERO HIGH BYTE.
        LXI     H,STADCT;FOR START ADD. OF CORRECTION FACTOR TABLE (02,01,00)
        DAD     D         ; ADD REV. COUNT.
        MOV     A,M       ; READ CORRECTION FACTOR
        CPI     01H       ; TEST FACTOR
        MOV     A,C       ; K(CC)*SIN(C)=K(C)*0.056*SIN(C)
        JZ      CRECT1    ; JUMP IF 01 [K(CC)=K(C)*0.028*SIN(C)]
        JNC     CRECT2    ; JUMP IF 02 [K(CC)=K(C)*0.056*SIN(C)]
        SUB     A         ;      IF 00 [K(CC)=K(C)*0.000*SIN(C)]
;
CRECT1: ANA     A         ; P FLAG SET IF SIN(C) POS
        JP      PLUS      ; JUMP IF POS.
        STC               ; SET CARRY PRIOR TO ROTATE RIGHT
PLUS:   RAR               ; DIVIDE BY TWO (ZERO δ UNAFFECTED BY DIVISION)
;
CRECT2: MOV     C,A       ; CORRECTION TERM IN C REG.
        LXI     H,STADCC;FOR START ADD. OF COS(C) TABLE
        POP     D         ; RECALL ANGLE COUNT WHEEL C.
        DAD     D         ; ADD TO START ADD.
        MOV     A,M       ; READ K(C)*COS(C)
        SUB     C         ; K(C)*[COS(C)-δ*SIN(C)]
        MOV     L,A       ; SAVE IN H,L REG. PAIR
        MVI     H,00H     ;
        CPI     080H      ; TEST FOR NEG.
        JC      POS8      ; JUMP IF POS.
        MVI     H,0FFH    ; 2s COMP IN UPPER BYTE.
;
POS8:   POP     B         ; RECALL RUNNING SUM
        DAD     B         ; K(A)*COS(A)+K(B)*COS(B)+K(C)[COS(C)-δ*SIN(C)]
        MOV     B,H       ; SAVE IN B,C REG.PAIR
        MOV     C,L
        LXI     H,WCD     ;FOR ANGLE COUNTER WHEEL D
        MOV     E,M       ; READ ANGLE LOW BYTE
        MVI     D,00H     ; ZERO HIGH BYTE
        LXI     H,STADD   ;FOR START ADD. COS(D)
        DAD     D         ; ADD ANGLE
        MOV     A,M       ; READ K(D)*COS(D)
        MOV     L,A       ; SAVE IN H,L REG. PAIR
        MVI     H,00H     ;
        CPI     080H      ; TEST FOR NEG.
        JC      POS9      ; JUMP IF POS.
        MVI     H,0FFH    ; 2s COMP IN UPPER BYTE
POS9:   DAD     B         ; K(A)*COS(A)+K(B)*COS(B)+K(C)*[COS(C)
                                    -δ*SIN(C)]+K(D)*COS(D)
;________________________________________________________________________
;
; SCALING ROUTINE:- MULTIPLY BY 64 decimal
;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        MOV     A,L
        RRC
        RRC
```

```
        MOV     C,A
        ANI     3FH
        MOV     B,A
        MOV     A,C
        ANI     0C0H
        MOV     C,A
        MOV     A,H
        RRC
        RRC
        ANI     0C0H
        ORA     B
        MOV     B,A       ; SCALED SUM IN B,C REG.PAIR
;_________________________________________________________________________
;
; DIVIDE SUM OF COSINES VALUE BY TIME^2 USING THE 8231 A.P.U. THIS EFFECTIVELY
; MULTIPLIES M.R. BY "OMEGA"^2 AFTER RESOLUTION INTO THE VERTICAL AXIS.  FINAL
; RESULT IS COMPUTED INSTANTANEOUS VALUE OF FORCE. FOR THE FIRST 12 DIVISIONS,
; A DEFAULT DIVISOR IS USED.
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
DIVIDE: LXI     H,TIMOUT;FOR TIME OF "E" PULSE PERIOD
        MOV     E,M       ; READ TIME
        MVI     L,0FH     ;FOR DIVISION COUNTER
        MOV     A,M       ; READ COUNT.
        CPI     0DH       ; TEST FOR 12
        JZ      RITIM     ; JUMP IF 12
        INR     M         ; INCREMENT DIVISION COUNTER.
        MVI     E,7FH     ; SET DEFAULT VALUE OF TIME DIVISOR.
;
RITIM:  LXI     H,APUD    ;FOR TOP OF A.P.U. STACK
        MOV     M,C       ; LOAD LOW BYTE OF DIVIDEND INTO STACK
        MOV     M,B       ; LOAD HIGH BYTE OF DIVIDEND INTO STACK
        MOV     M,E       ; LOAD LOW BYTE OF "TIME" DIVISOR INTO STACK
        MVI     M,00H     ; LOAD HIGH BYTE OF "TIME" DIVISOR INTO STACK
        INR     L         ;FOR A.P.U. COMMAND
        MVI     M,6FH     ; SET 01101111 ≡NO SVREQ≡SINGLE PRECISION≡FIXED PT≡DIVIDE≡
        MVI     B,0FFH    ; LOAD CYCLE COUNTER
DEL:    MOV     A,M       ; READ STATUS OF A.P.U.
        DCR     B         ; DECREMENT CYCLE COUNTER
        JZ      ENDE      ; JUMP OUT IF  A.P.U. STUCK
        RLC
        JC      DEL       ; JUMP BACK IF A.P.U. BUSY
;
        DCR     L         ;FOR TOP OF A.P.U. STACK
        MOV     A,M       ; READ UPPER BYTE OF DIVIDEND
        RAL               ; SET CARRY IF NEGATIVE
        MOV     A,M       ; READ LOWER BYTE OF DIVIDEND
        RAR               ; DIVIDE BY 2 (CARRY PRESERVES 2's COMP NEG)
        MVI     M,00H     ; LOAD LOWER BYTE OF DIVIDEND MULTIPLIED BY 256
        MOV     M,A       ; LOAD UPPER BYTE OF DIVIDEND MULTIPLIED BY 256
        MOV     M,E       ; LOAD LOWER BYTE OF "TIME" DIVISOR INTO STACK
        MVI     M,00H     ; LOAD UPPER BYTE OF "TIME" DIVISOR INTO STACK.
        INR     L         ;FOR A.P.U. COMMAND
        MVI     M,6FH     ; SET 01101111 ≡NO SVREQ≡SINGLE PRECISION≡FIXED PT≡DIVIDE≡
        MVI     B,0FFH    ; LOAD CYCLE COUNTER
DEL2:   MOV     A,M       ; READ STATUS OF A.P.U.
        DCR     B         ; DECREMENT CYCLE COUNTER
        JZ      ENDE      ; JUMP OUT IF C.P.U. STUCK
        RLC
        JC      DEL2      ; JUMP BACK IF A.P.U. BUSY
;
```

```
        DCR     L       ;FOR TOP OF A.P.U. STACK
        MOV     A,M     ; READ UPPER BYTE AND DISCARD
        RAL             ; SET CARRY IF NEGATIVE
        MOV     A,M     ; READ LOWER BYTE
        RAR             ; DIVIDE BY 2 (CARRY PRESERVES 2's COMP NEG.)
        ADI     80H     ; CONVERT FROM OFFSET BINARY TO BINARY.
;-----------------------------------------------------------------
;
;  OUTPUT COMPUTED FORCE TO D/A CONVERTER.
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
        LXI     H,PRTA  ;FOR PORT A.
        MOV     M,A     ; OUTPUT INCREMENTAL VALUE OF FORCE
;---------------------------------------------------------------------
;
; INCREMENT WHEEL ANGLE COUNTERS A, B, C & D. UNMASK INTERRUPTS A, B, C &
; D AT SPECIFIED ANGLE COUNTS.  DETECT LOSS OF A OR B INTERRUPTS & ABORT.
; DETECT & REMOVE OVERCOUNT IN C & D WHEEL ANGLE COUNTERS .
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
ENDE:   LXI     H,WCAL  ;FOR ANGLE COUNTER WHEEL A LOW BYTE
        MOV     A,M     ; READ
        INR     A       ; INCREMENT
        MOV     M,A     ; STORE
        JNZ     POS10   ; JUMP IF LOW BYTE < 256 dec
        INX     H       ;FOR ANGLE COUNTER WHEEL A UPPER BYTE
        INR     M       ; INCREMENT
        LXI     H,PICB  ;FOR P.I.C. (OCW1)
        MOV     A,M     ; READ MASK REG
        ANI     0EFH    ; UNMASK 11101111 IRA
        MOV     M,A     ; SET MASK REG
POS10:  LXI     H,WCBL  ;FOR ANGLE COUNTER WHEEL B LOW BYTE
        MOV     A,M     ; READ
        INR     A       ; INCREMENT
        MOV     M,A     ; STORE
        JNZ     POS16   ; JUMP IF LOW BYTE < 256
        INX     H       ;FOR ANGLE COUNTER WHEEL B UPPER BYTE
        INR     M       ; INCREMENT
        MOV     A,M     ; WCBU COUNT IN ACC
        LXI     H,WCAU  ;FOR ANGLE COUNTER WHEEL A UPPER BYTE
        ADD     M       ; ADD WCAU COUNT
        CPI     03H     ; COUNT OF 3?
        JC      POS13   ; CONTINUE FORCE COMPUTATION IF COUNT < 3
        LXI     H,MSP   ;FOR MOTOR SPEED POINTER (ABORT IF COUNT = OR > 3)
        INR     M       ; INCREMENT
        JMP     ABORT   ; ABORT IF A OR B PULSE FAILS
POS13:  LXI     H,PICB  ;FOR P.I.C. (OCW1)
        MOV     A,M     ; READ MASK REG
        ANI     0DFH    ; UNMASK 11011111 IRB
        MOV     M,A     ; SET MASK REG
POS16:  LXI     H,WCC   ;FOR ANGLE COUNTER WHEEL C
        MOV     A,M     ; READ COUNTER
        CPI     5CH     ; COUNT OF 92?
        JM      POS5    ; JUMP IF LESS
        MVI     M,59H   ; SET COUNTER TO 89
POS5:   INR     M
        CPI     58H     ; COUNT OF 88?
        JM      POS17   ; JUMP IF LESS
        LXI     H,PICB  ;FOR P.I.C. (OCW1)
        MOV     A,M     ; READ MASK REG
```

```
        ANI     0F7H    ; UNMASK 11110111 IRC
        MOV     M,A     ; SET MASK REG
POS17:  LXI     H,WCD   ;FOR ANGLE COUNTER WHEEL D
        MOV     A,M     ; READ COUNTER
        CPI     5CH     ; COUNT OF 92?
        JM      POS6    ; JUMP IF LESS
        MVI     M,59H   ; SET COUNTER TO 89
POS6:   INR     M
        CPI     58H     ; COUNT OF 88?
        JM      POS18   ; JUMP IF LESS
        LXI     H,PICB  ;FOR P.I.C. (OCW1)
        MOV     A.M     ; READ MASK REG
        ANI     0FBH    ; UNMASK 11111011 IRD
        MOV     M,A     ; SET MASK REG
;
POS18:  RET             ; RETURN TO EXEC
;________________________________________________________________________
;
; --INT. SERVICE ROUTINE "WHEEL A".--
; SET WHEEL A ANGLE COUNTER TO 1.  INCREMENT MOTOR SPEED POINTER, FETCH MOTOR
; SPEED FROM TABLE & OUTPUT SPEED DEMAND.  COMPENSATE WHEEL D REV COUNTER FOR
; PREVIOUS OVERRUN.  AFTER 8 REVS OF WHEEL A UNMASK "COINCIDENCE", AFTER EACH
; REV MASK "WHEEL A". OUTPUT 1 microSEC PULSE ACKNOWLEDGE A.
; OUTPUTS: MOTOR SPEED
; INPUTS: FROM MOTOR SPEED TABLE
; INTERRUPT MASKS: ISRA MASKED, ISRCI UNMASKED ON 8th REV
;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
ISRA:   LXI     H,0001H ; SET 1
        SHLD    WCAL    ;IN ANGLE COUNTER WHEEL A
        LXI     H,MSP   ;FOR MOTOR SPEED POINTER
        MOV     E,M     ; LOW BYTE IN REG.E
        MVI     D,00H   ; ZERO HIGH BYTE IN REG.D
        INR     M       ; INC. FOR REV. OF WHEEL A.
        MVI     A,02H   ; SECOND ISRA
        SUB     E       ; CHECK
        JNZ     CONT1   ; JUMP IF NOT
        LXI     H,WCBL  ;FOR ANGLE COUNTER WHEEL B
        MOV     A,M     ; READ
        CPI     08H     ; 7 "E" PERIODS PLUS 1
        JC      CONT2   ; JUMP IF LESS
        INR     A
CONT2:  LXI     H,RCNT  ;FOR REV COUNTER
        MOV     M,A     ; INITIALISE REV COUNTER
;
CONT1:  LXI     H,SPEED ;FOR MOTOR SPEED TABLE START ADDR.
        DAD     D       ; ADD M.S.P. COUNT TO START ADDRESS
        MOV     A,M     ; READ DEMAND SPEED
        LXI     H,PRTB  ;FOR PORT B
        MOV     M,A     ; OUTPUT SPEED
        LXI     H,PORTC ;FOR PORT C OUTPUT DATA
        MOV     A,M     ; READ
        ANI     0F7H    ; REMOVE BIT 3 (ACKA)
        LXI     H,PRTC  ;FOR PORT C
        MOV     M,A     ; START ACKA VIA PORT C
        ORI     08H     ; REPLACE BIT 3
        MOV     M,A     ; END ACKA VIA PORT C
        LXI     H,PICB  ;FOR P.I.C. OCW1
        MOV     A,M     ; READ MASK REG
        ORI     10H     ; MASK 00010000 IRA
        MOV     M,A     ; SET MASK REG.
```

```
        MOV     A,E      ; REV COUNT INTO ACC.
        CPI     08H      ; TEST 8 REVOLUTIONS OF WHEEL A.
        JNZ     POS21
        MOV     A,M      ; READ MASK REG
        ANI     0FEH     ; UNMASK 11111110 IRCI
        MOV     M,A      ; SET MASK REG
;
POS21:  RET
;________________________________________________________________________
;
; --INT. SERVICE ROUTINE "WHEEL A".-- (DURING INITIALISATION ONLY)
; SET WHEEL A ANGLE COUNTER TO 1.  INCREMENT MOTOR SPEED POINTER.  AFTER
; EACH REV MASK "WHEEL A". SEND 1 microSEC PULSE ACKNOWLEDGE A1.
; OUTPUTS: NONE
; INPUTS: NONE
; INTERRUPT MASKS: ISRA MASKED
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
ISRA1:  PUSH    H
        LXI     H,0001H ; SET 1
        SHLD    WCAL     ;IN ANGLE COUNTER WHEEL A
        LXI     H,MSP    ;FOR MOTOR SPEED POINTER
        INR     M        ; INCREMENT M.S.P.
        LXI     H,PICB   ;FOR P.I.C. (OCW1)
        MOV     A,M      ; READ
        ORI     10H      ; MASK 00010000 IRA
        MOV     M,A
        LXI     H,PORTC ;FOR PORT C OUTPUT DATA
        MOV     A,M      ; READ
        ANI     0F7H     ; REMOVE BIT 3 ACKA
        LXI     H,PRTC   ;FOR PORT C
        MOV     M,A      ; START ACKA VIA PORT C
        ORI     08H      ; REPLACE BIT 3
        MOV     M,A      ; END ACKA VIA PORT C
        POP     H
        RET
;
;________________________________________________________________________
;
; --INT. SERVICE ROUTINE "WHEEL B".--
; SET ANGLE COUNTER WHEEL B TO 1.  AFTER EACH REV MASK "WHEEL B".
; INPUTS: NONE
; OUTPUTS: NONE
; INTERRUPT MASKS: ISRB MASKED
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
ISRB:   LXI     H,0001H ; SET 1
        SHLD    WCBL     ; IN ANGLE COUNTER WHEEL B.
        LXI     H,PICB   ;FOR P.I.C. OCW1
        MOV     A,M      ; READ MASK REG
        ORI     20H      ; MASK 00100000 IRB
        MOV     M,A      ; SET MASK REG
;
        RET
```

```
;______________________________________________________________________
;
; --INT. SERVICE ROUTINE "WHEEL C".--
; SET WHEEL C ANGLE COUNTER TO 1.  INCREMENT WHEEL C REV COUNTER & SET TO
; 1 EVERY 19th REV.  AFTER EACH REV MASK "WHEEL C".
; INPUTS: NONE
; OUTPUTS: NONE
; INTERRUPT MASKS: ISRC MASKED
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
ISRC:   LXI     H,WCC   ;FOR ANGLE COUNTER WHEEL C.
        MVI     M,01H   ; SET 1
        LXI     H,RCNT  ;FOR REV. COUNTER.
        MVI     A,13H   ; 19 REVS
        SUB     M       ; TEST
        JNZ     POS11   ; JUMP IF LESS THAN 19
        MVI     M,00H   ; ZERO REV. COUNTER
POS11:  INR     M       ; INCREMENT REV. COUNTER
        LXI     H,PICB  ;FOR P.I.C. OCW1
        MOV     A,M     ; READ MASK REG.
        ORI     08H     ; MASK 00001000 IRC
        MOV     M,A     ; SET MASK REG
;
        RET
;______________________________________________________________________
;
; --INT. SERVICE ROUTINE "WHEEL D".--
; SET WHEEL D ANGLE COUNTER TO 1. AFTER EACH REV MASK "WHEEL D".
; INPUTS: NONE
; OUTPUTS: NONE
; INTERRUPT MASKS: ISRD MASKED.
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
ISRD:   LXI     H,WCD   ;FOR ANGLE COUNTER WHEEL D.
        MVI     M,01H   ; SET 1
        LXI     H,PICB  ;FOR P.I.C. OCW1
        MOV     A,M     ; READ MASK REG.
        ORI     04H     ; MASK 00000100 IRD
        MOV     M,A     ; SET MASK REG.
        RET
;______________________________________________________________________
; --INT. SERVICE ROUTINE "LOSS OF E PULSES"--
; COUNT OCCURRENCES OF L.E.P.  IF MORE THAN 80 THEN INCREMENT MOTOR SPEED
; POINTER & JUMP TO ABORT ROUTINE
; INPUTS: NONE
; OUTPUTS: NONE
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
LEP2:   LXI     H,LEPC  ;FOR LEP COUNTER
        INR     M       ; INCREMENT
        MOV     A,M     ; READ
        CPI     50H     ; 80 OCCURRANCES
        RC              ; RETURN IF LESS
        LXI     H,MSP   ;FOR MOTOR SPEED POINTER
        INR     M       ; INCREMENT M.S.P.
```

```
;_______________________________________________________________________
; --"ABORT" SUBROUTINE--
; IF MOTOR SPEED POINTER EXCEEDS COUNT OF 2, STOP MOTOR, EXTINGUISH RUN &
; INIT LAMPS & ILLUMINATE ABORT. ALL INTERRUPTS MASKED.
; INPUTS: NONE
; OUTPUTS: NONE
; INTERRUPT MASKS: ALL INTS MASKED.
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
ABORT:  LXI     H,MSP   ;FOR MOTOR SPEED POINTER
        MOV     A,M     ; READ
        CPI     03H     ; 2 REVS?
        JC      POS22   ; JUMP IF LESS
        LXI     H.PRTC  ;FOR PORT C
        MVI     M,0FBH  ; SET 11101 ≡P.I.T. G2 HIGH≡ABORT ON≡ INIT OFF≡
        LXI     H.PICB  ;FOR P.I.C. (OCW1)                     MOTOR STOP≡
        MVI     M,0FFH  ; MASK ALL INTERRUPTS
POS22:  RET
;_______________________________________________________________
;  SUBROUTINE CALL LISTS.
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

        ORG     07C0H
;
        JMP     ISRCI
        NOP
        JMP     ISRS
        NOP
        JMP     ISRD
        NOP
        JMP     ISRC
        NOP
        JMP     ISRA1
        NOP
        JMP     ISRB
        NOP
        JMP     ISRE
        NOP
        RET
        NOP
;______________________________________________________
        ORG     07E0H
;
        JMP     ISRCI
        NOP
        JMP     ISRS
        NOP
        JMP     ISRD
        NOP
        JMP     ISRC
        NOP
        JMP     ISRA
        NOP
        JMP     ISRB
        NOP
        JMP     ISRE
        NOP
        RET
;======================================================
        END
```

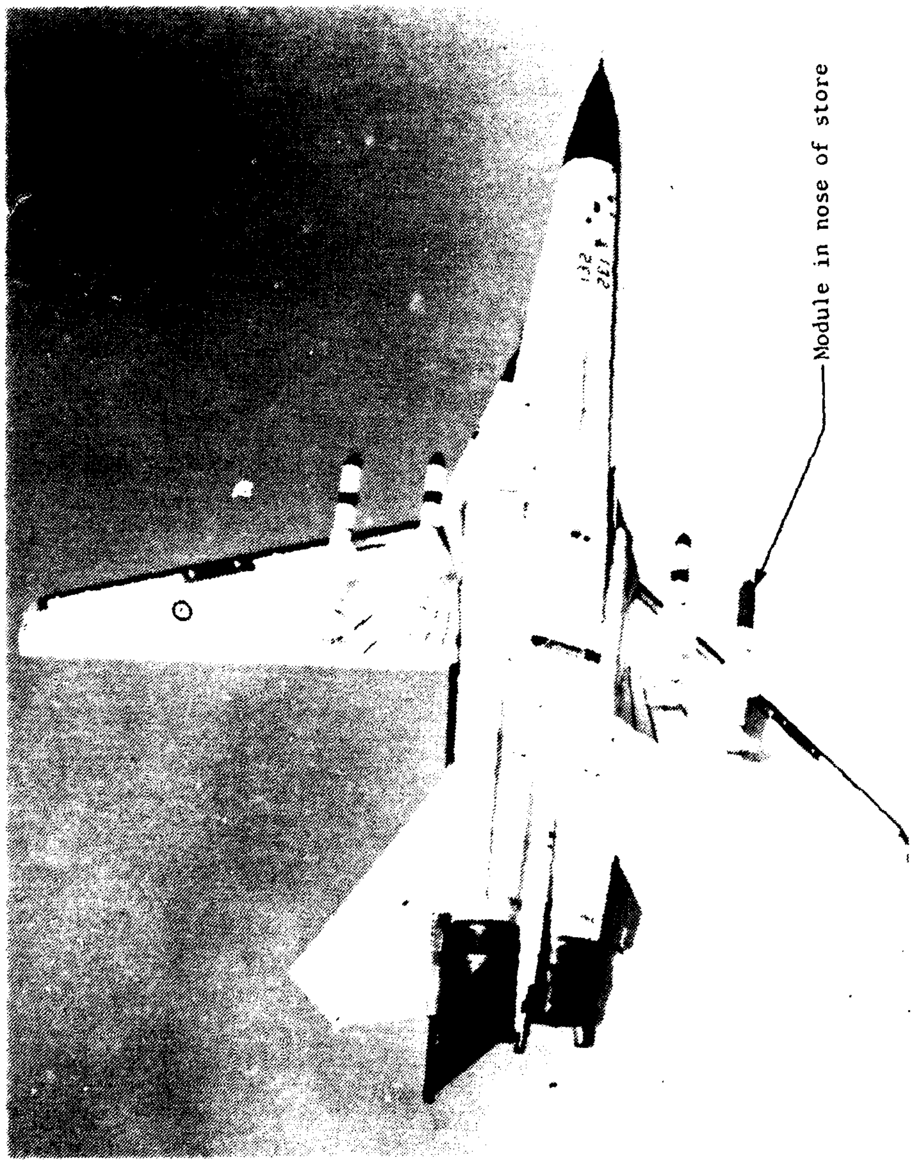


Figure 1. F111 Aircraft carrying store with exciter module

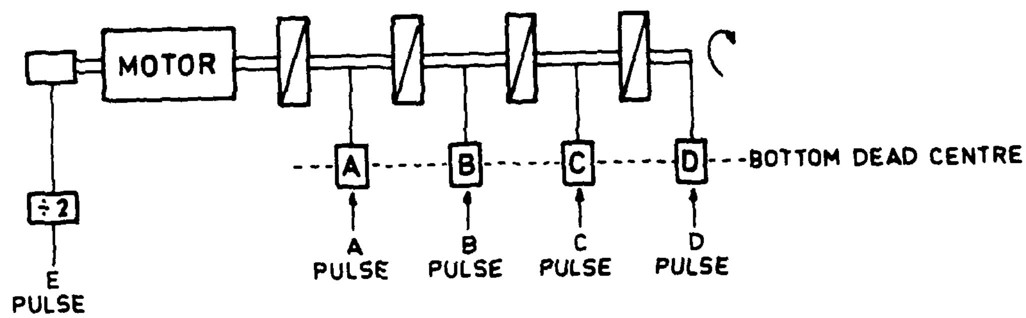


Figure 2. Mechanical load

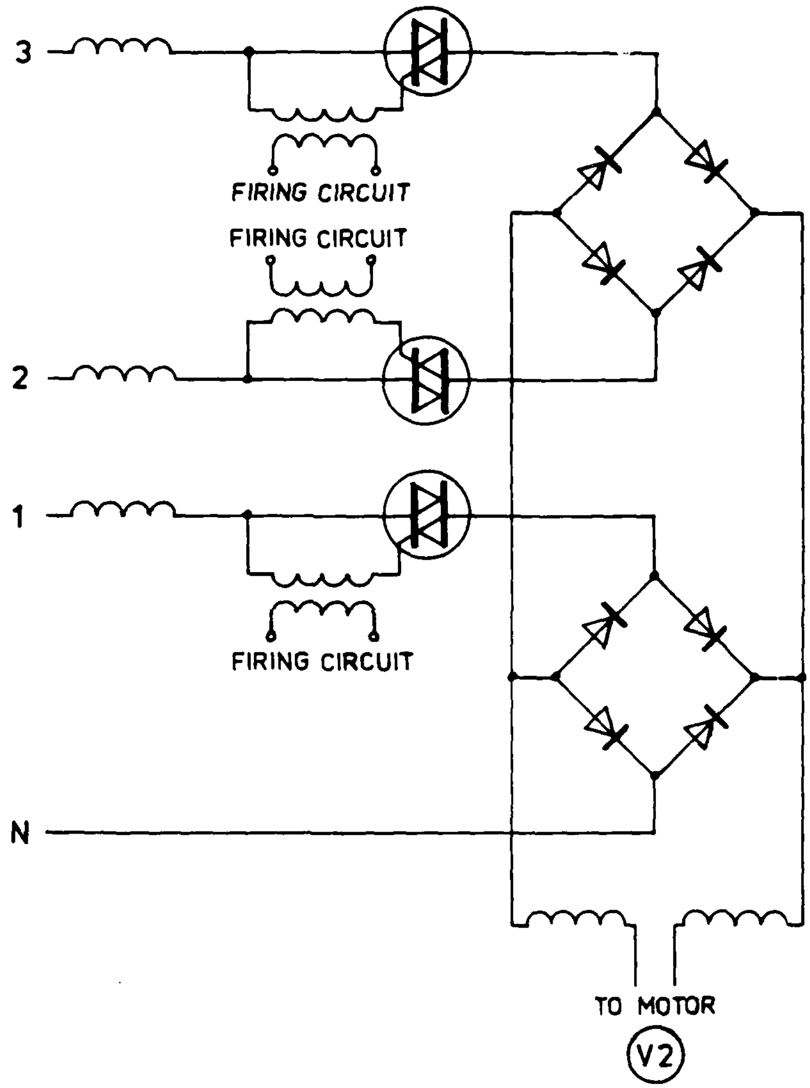


Figure 3. Rectifier and triac

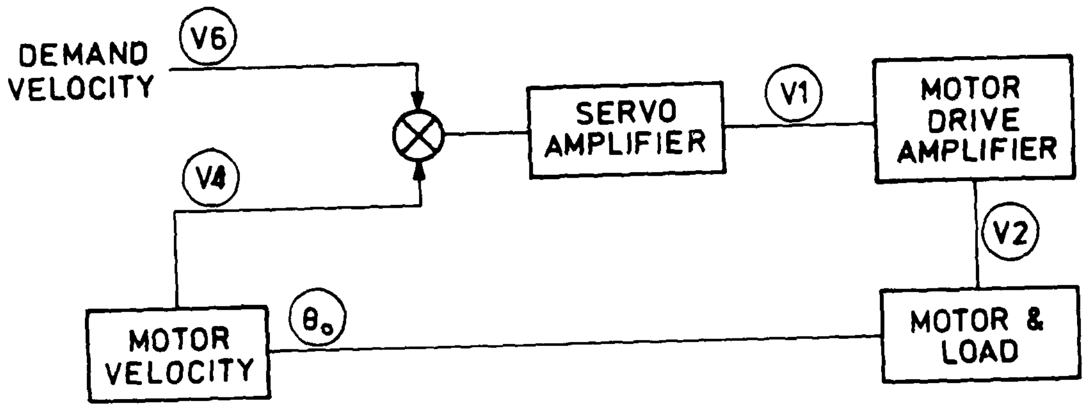


Figure 4. Overall system

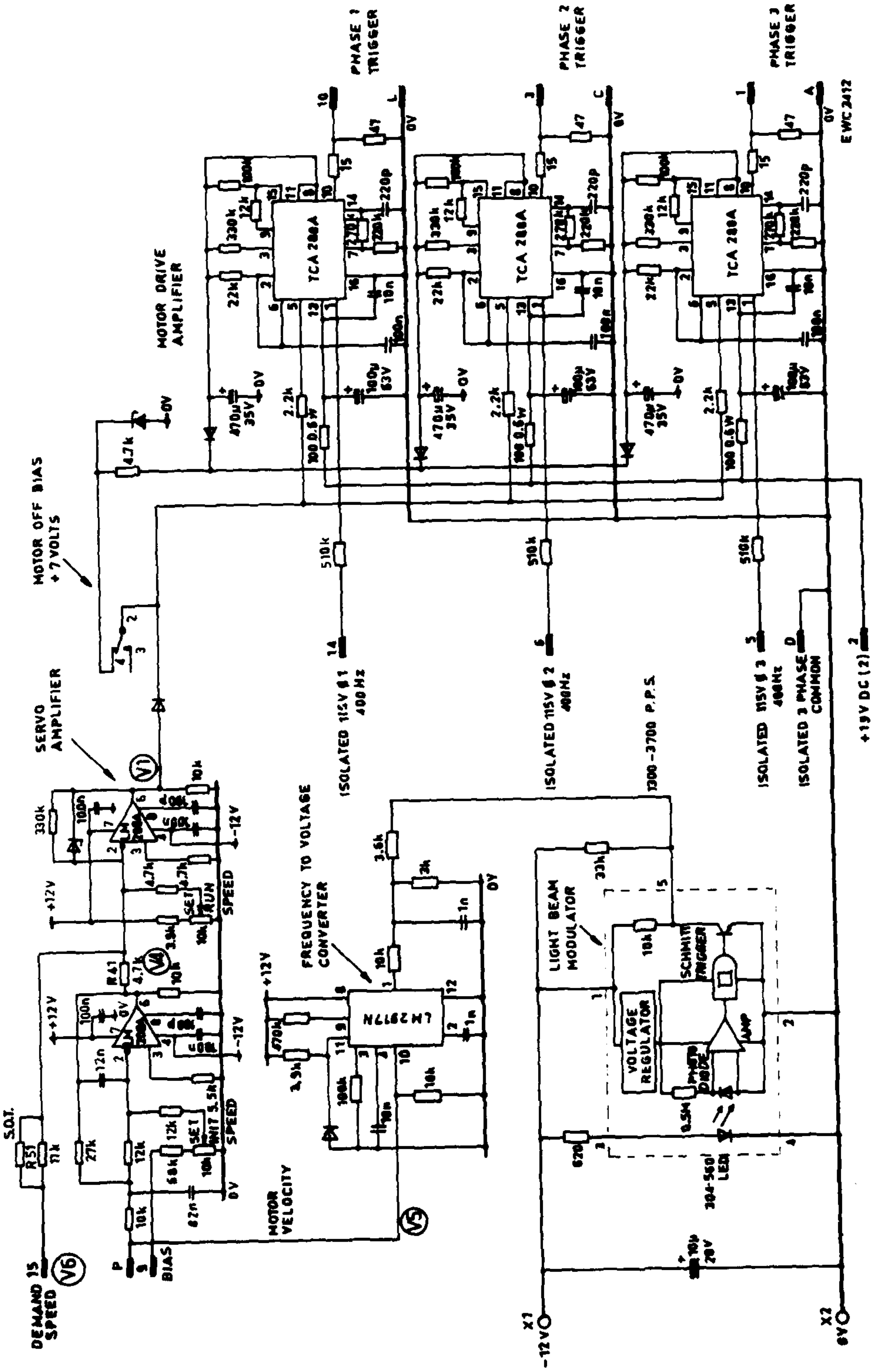


Figure 5. Velocity control

AEL-0242-TM
Figure 6

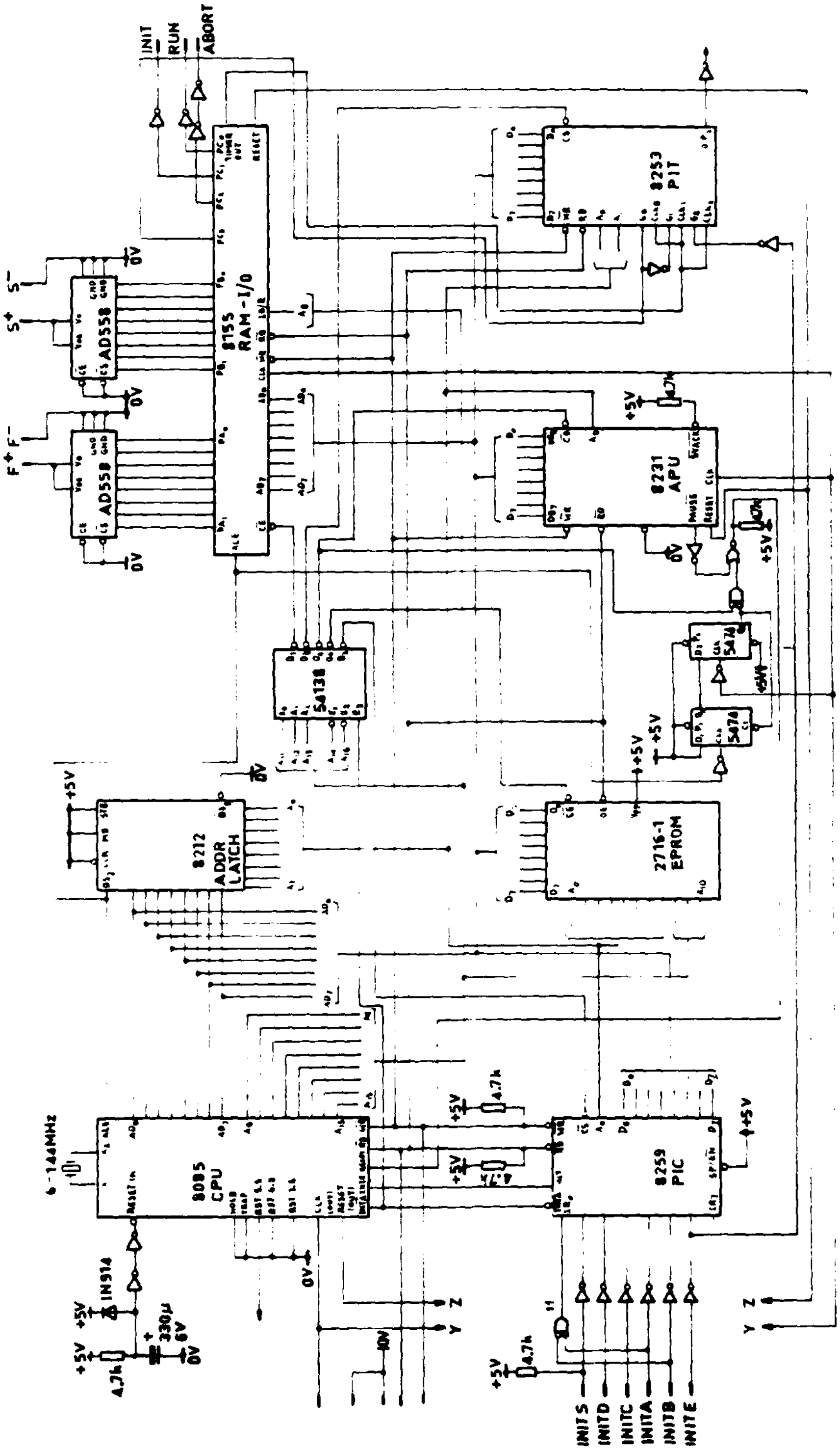


Figure 6. Microprocessor

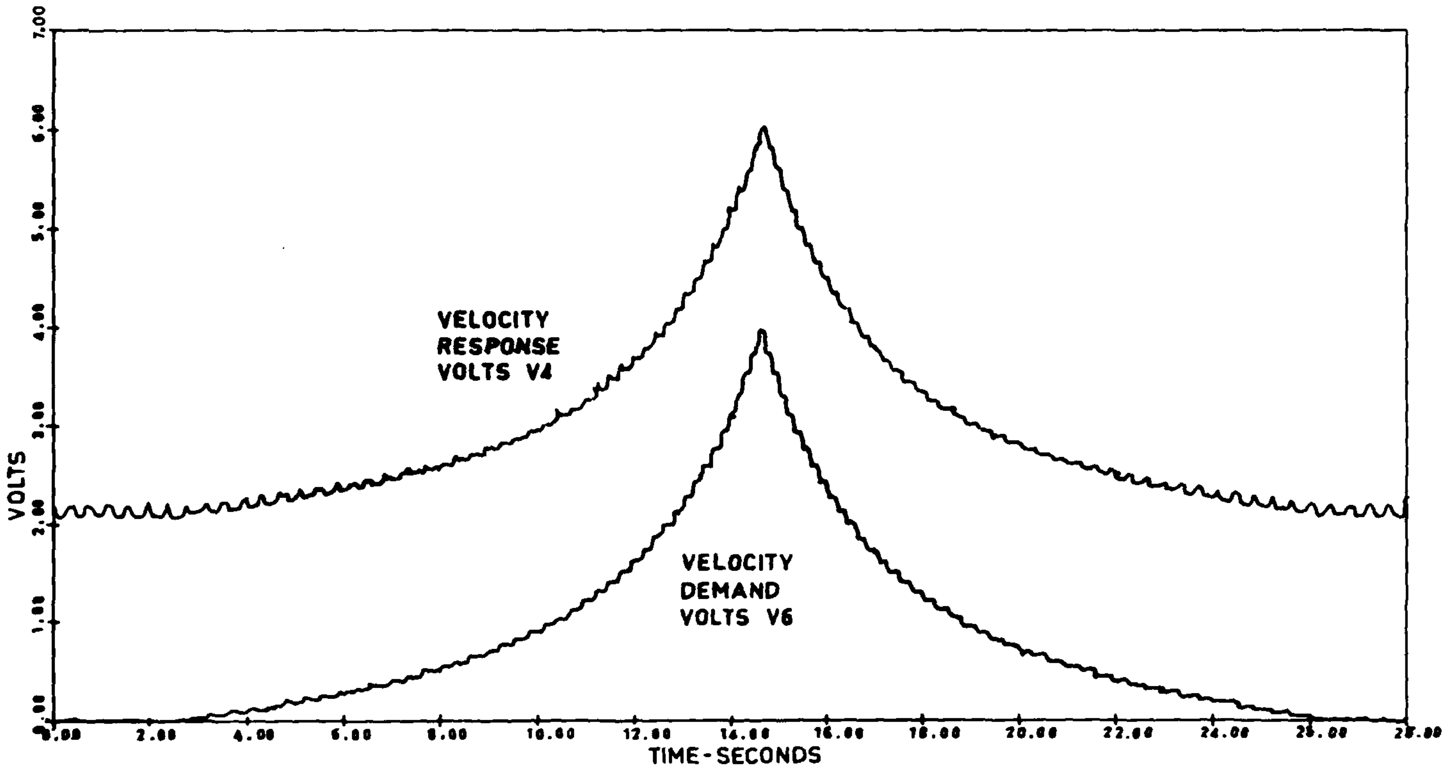


Figure 7. Operation in 1 'g' environment

AEL-0242-TM
Figure 8

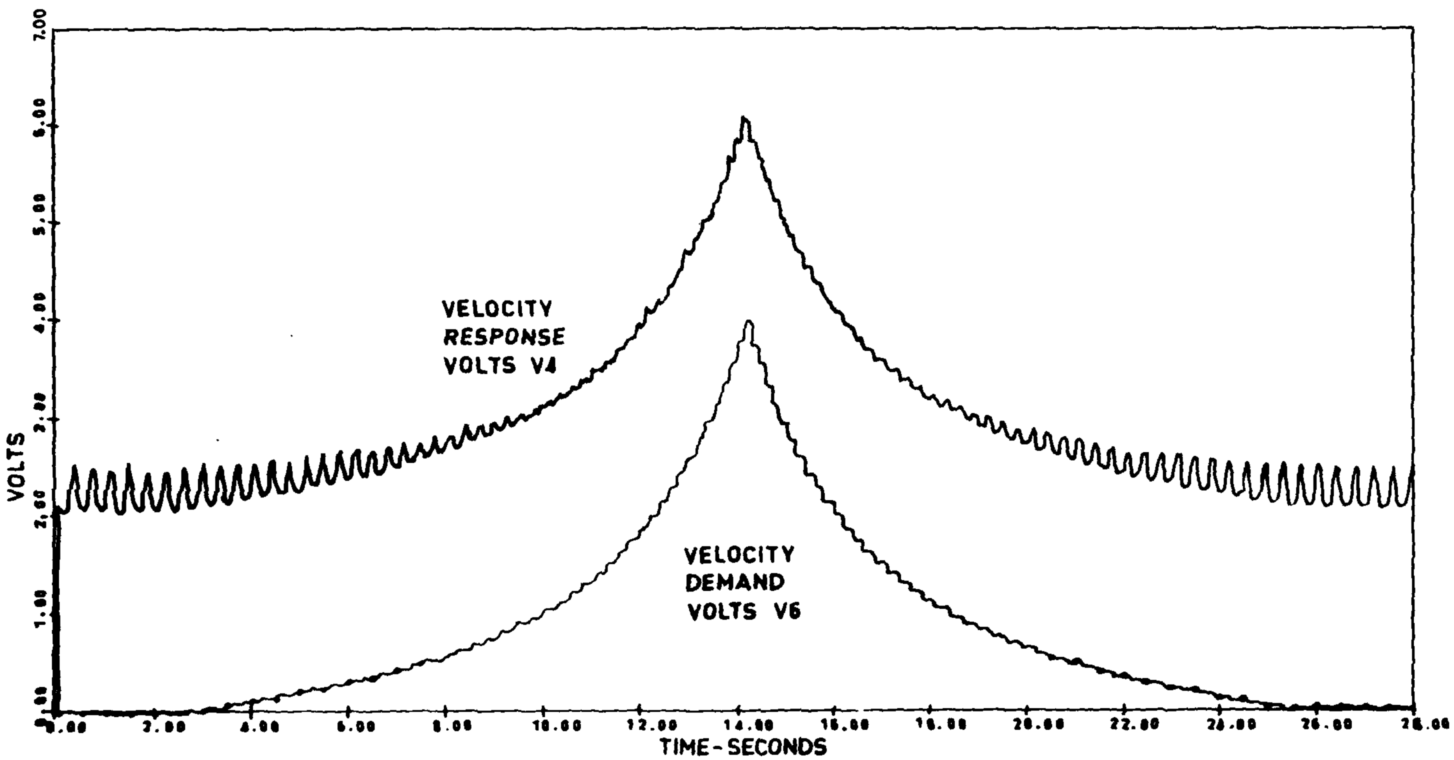


Figure 8. Operation in 2 'g' environment

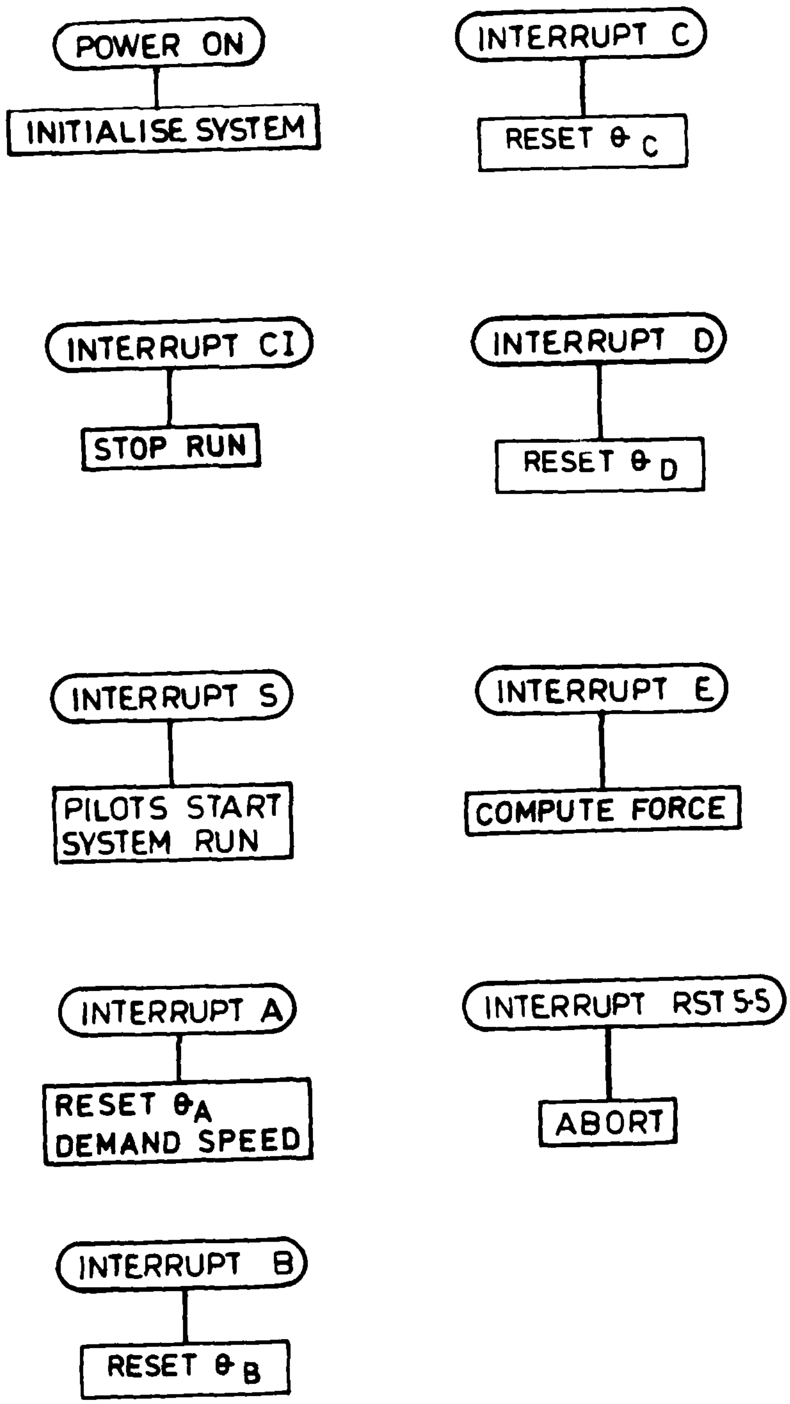


Figure 9. Software structure

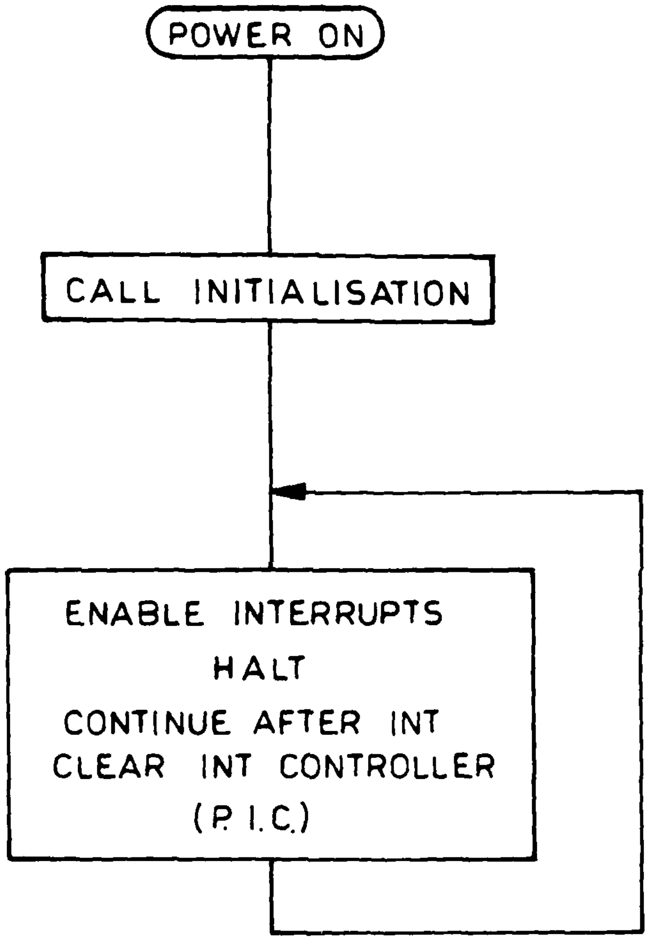


Figure 10. Executive programme

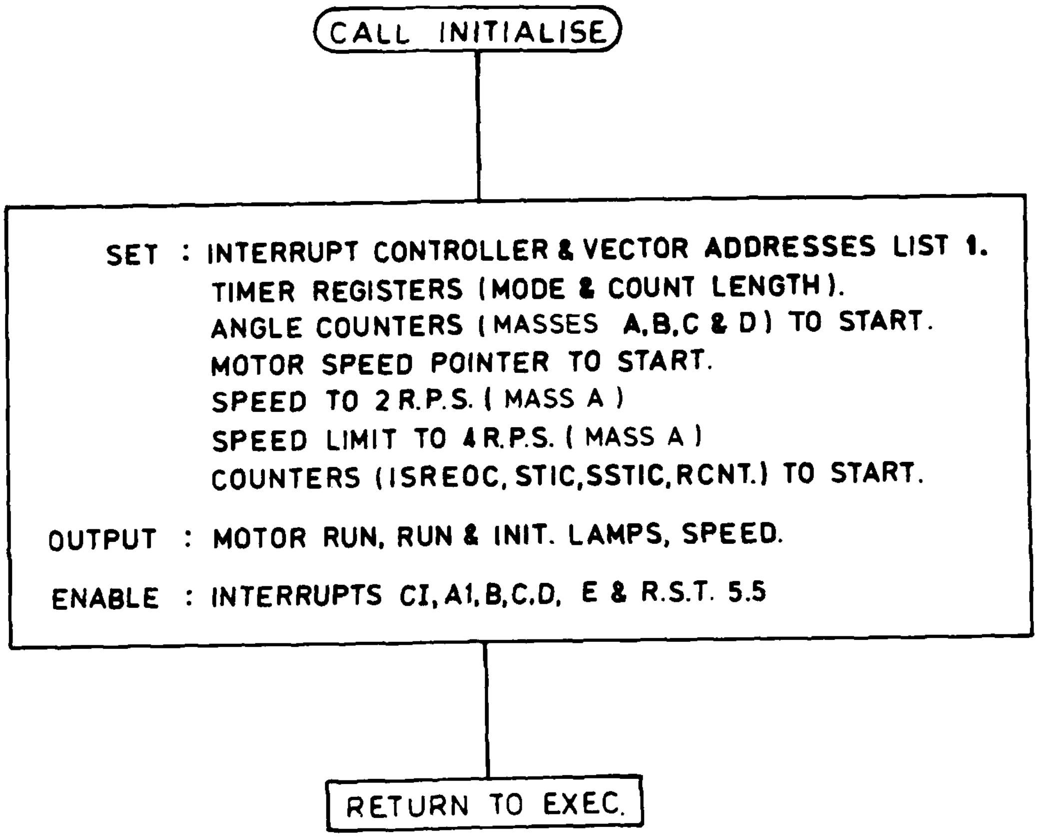


Figure 11. Initialisation subroutine

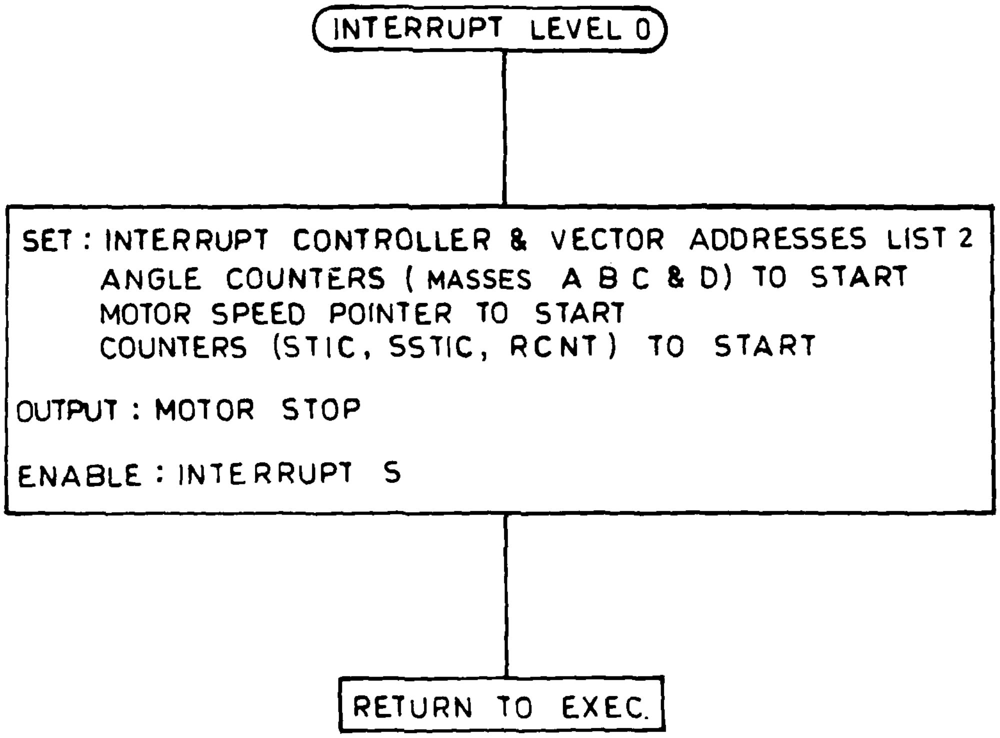


Figure 12. Coincidence subroutine

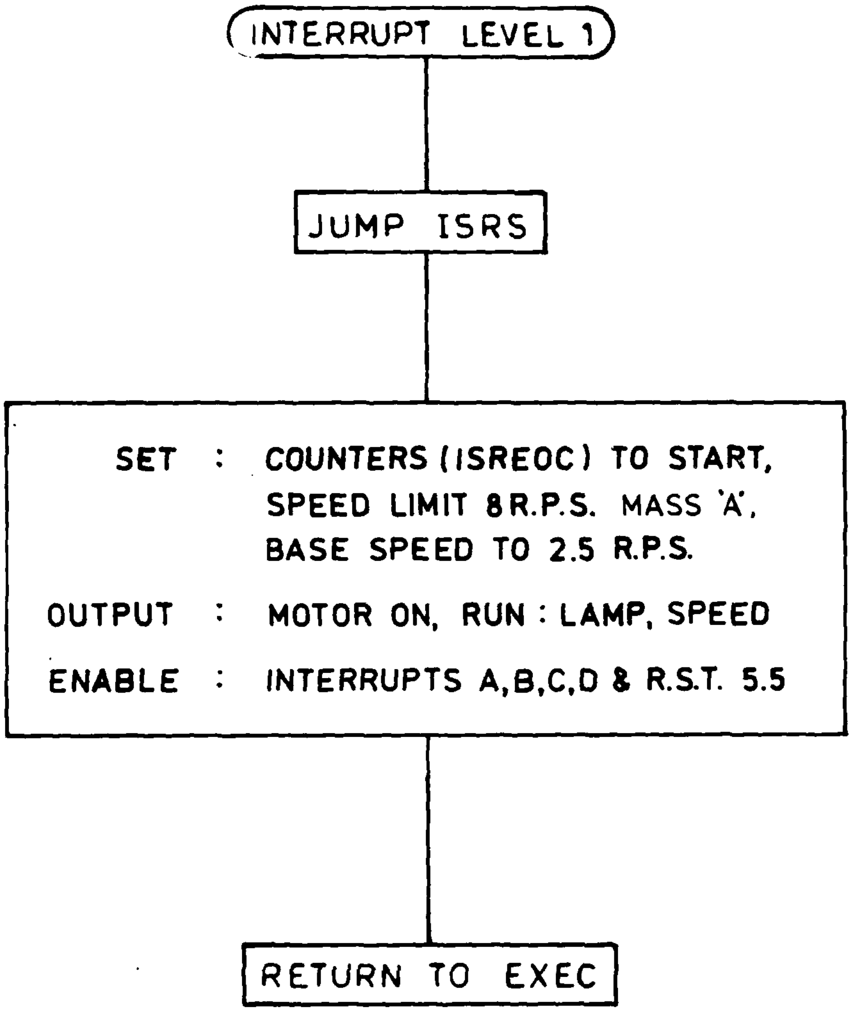


Figure 13. Pilot's start subroutine

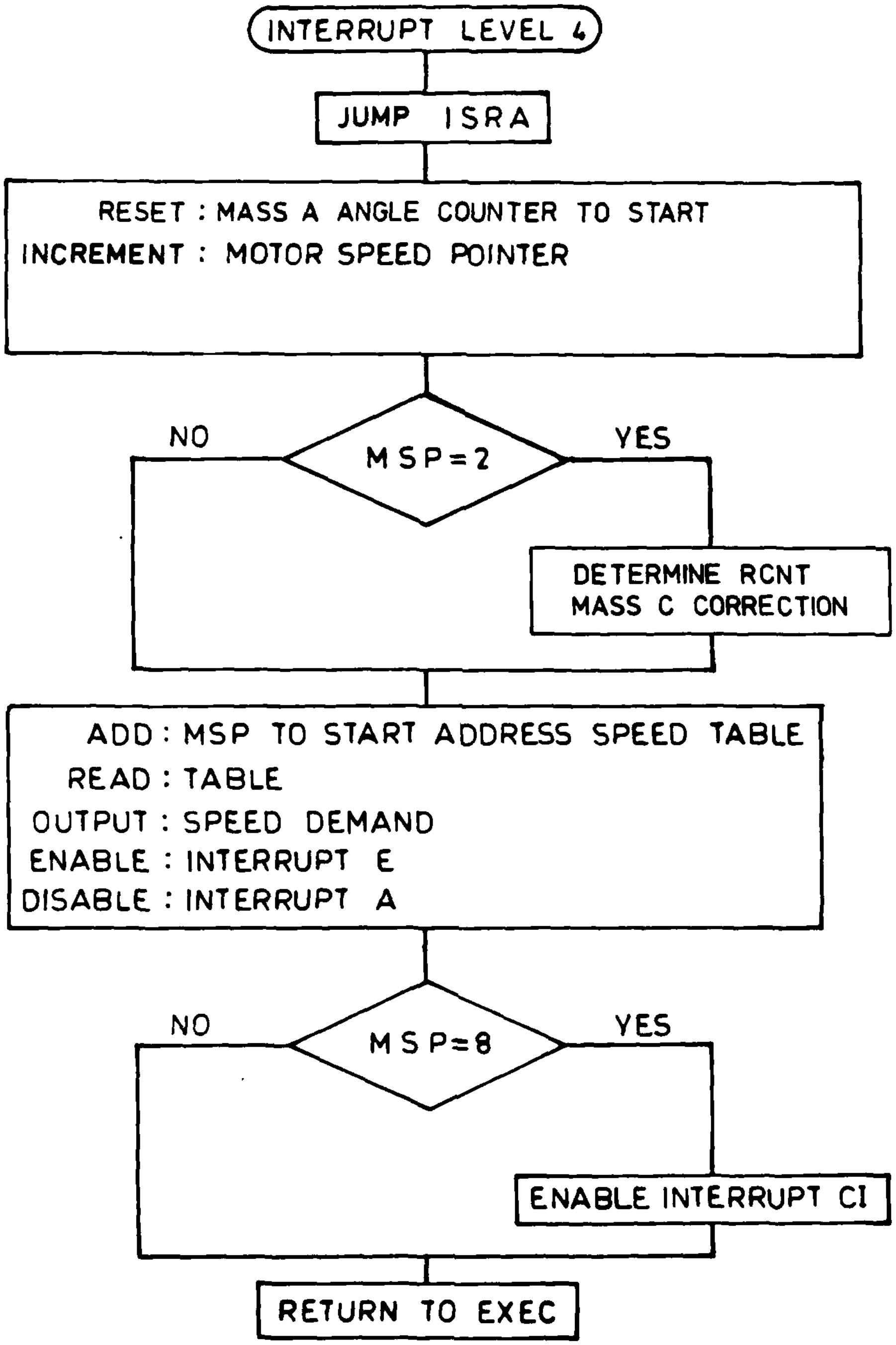


Figure 14(a). Wheel 'A' subroutine A (list 2)

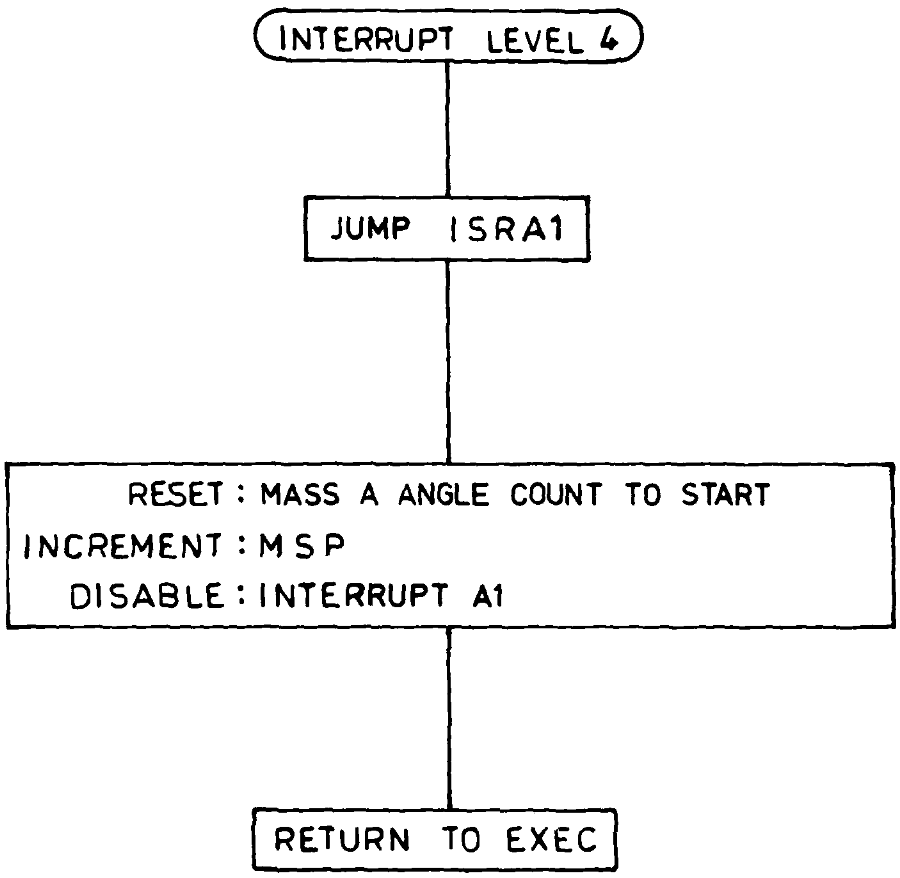


Figure 14(b). Wheel 'A' subroutine A1 (list 1)

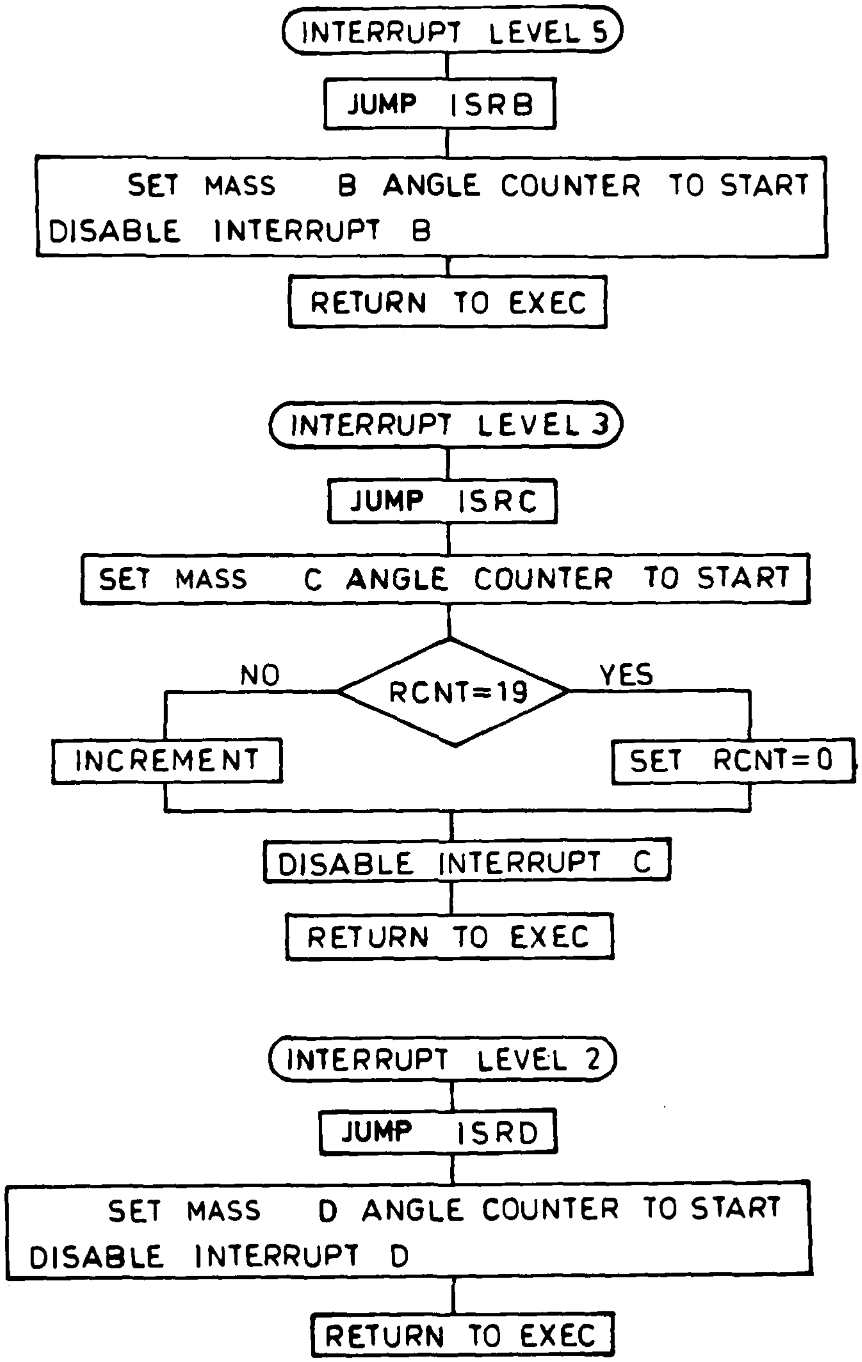


Figure 15. Wheels B, C and D subroutines B, C and D

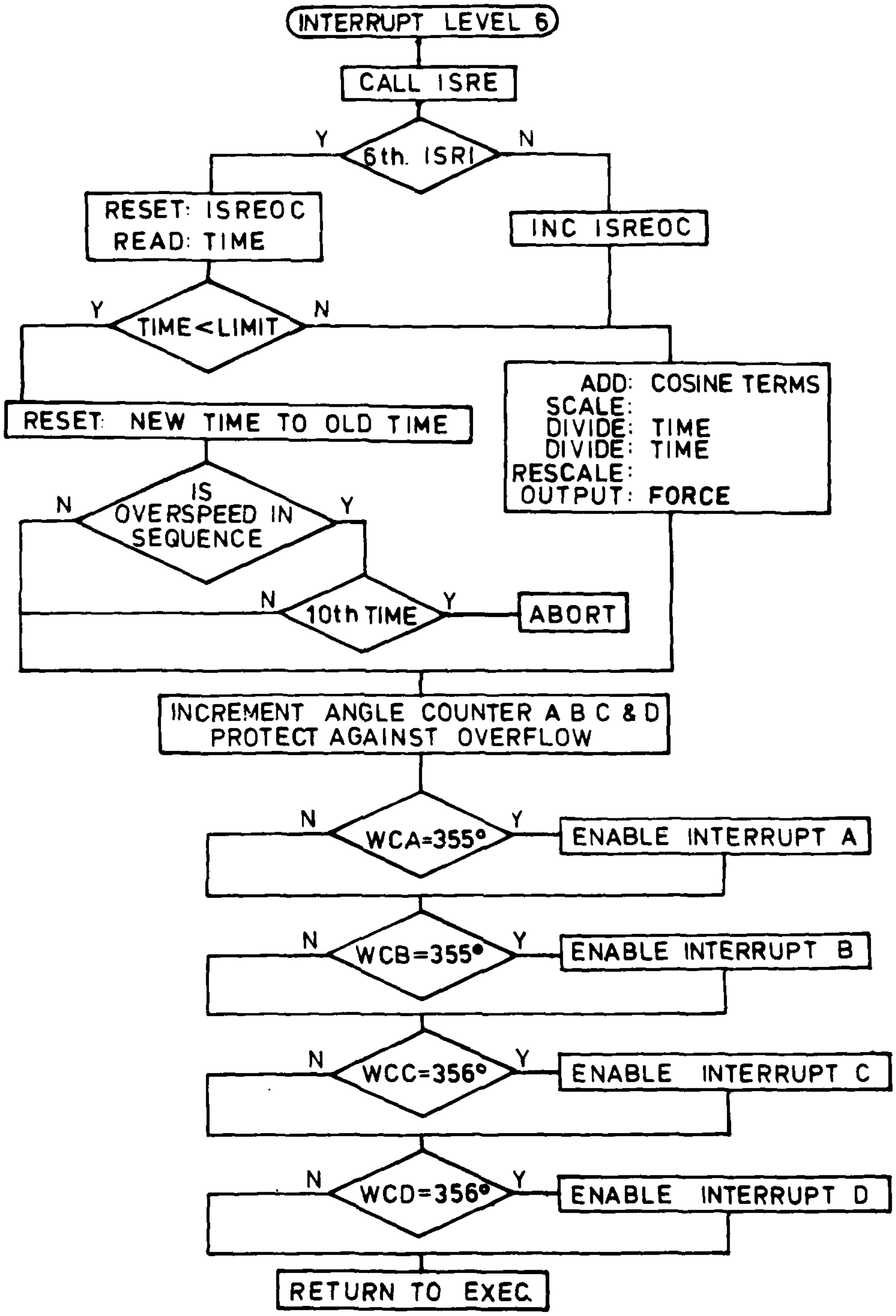


Figure 16. 'E' pulse subroutine

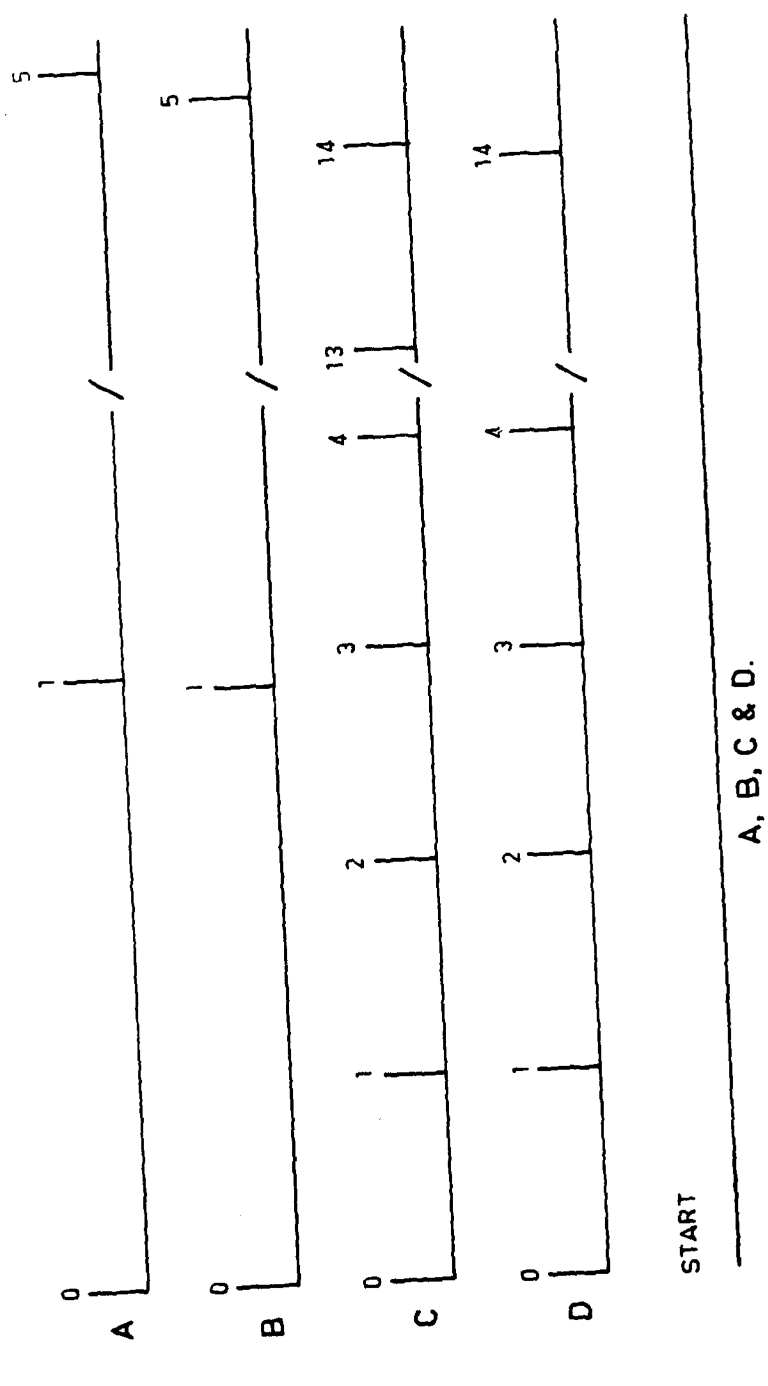


Figure 17. Interrupt pulse timing at run start

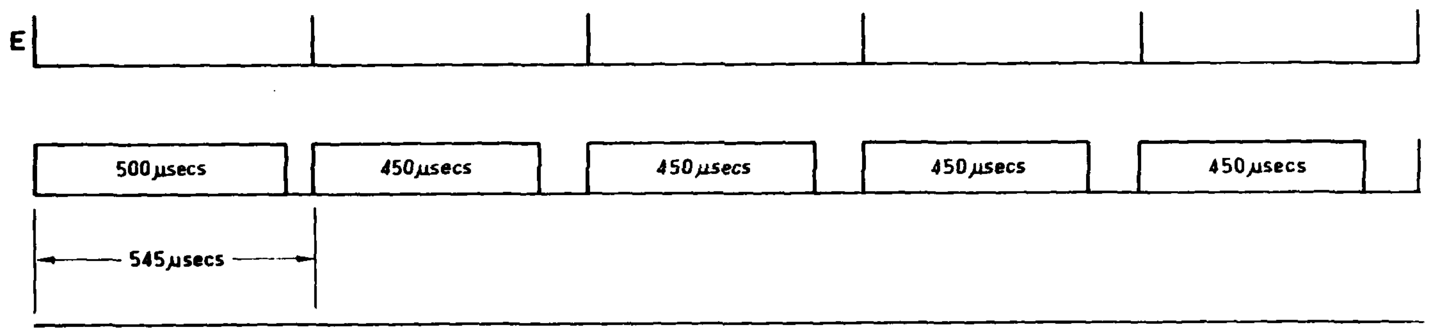


Figure 18. 'E' pulse timing and ISRE execution times at maximum speed

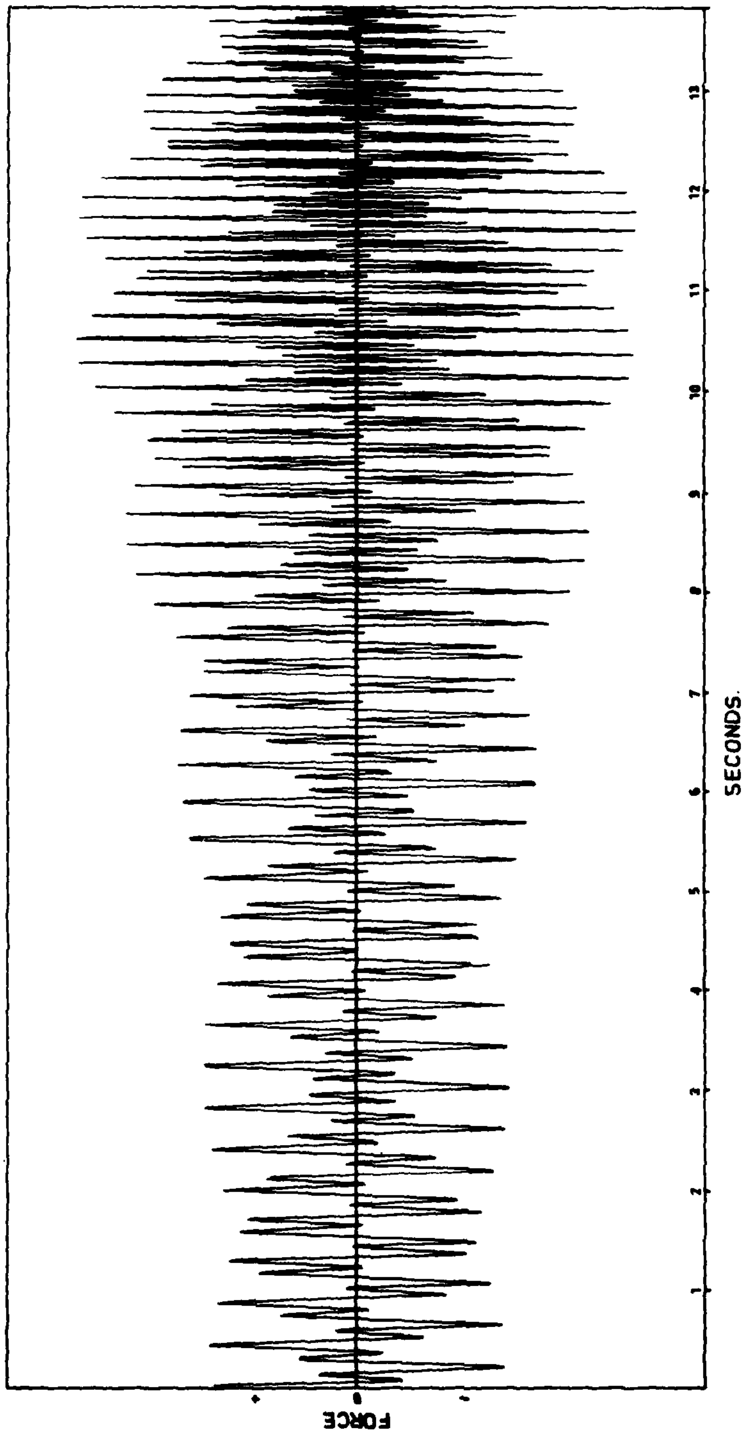


Figure 19. Excitation force waveform

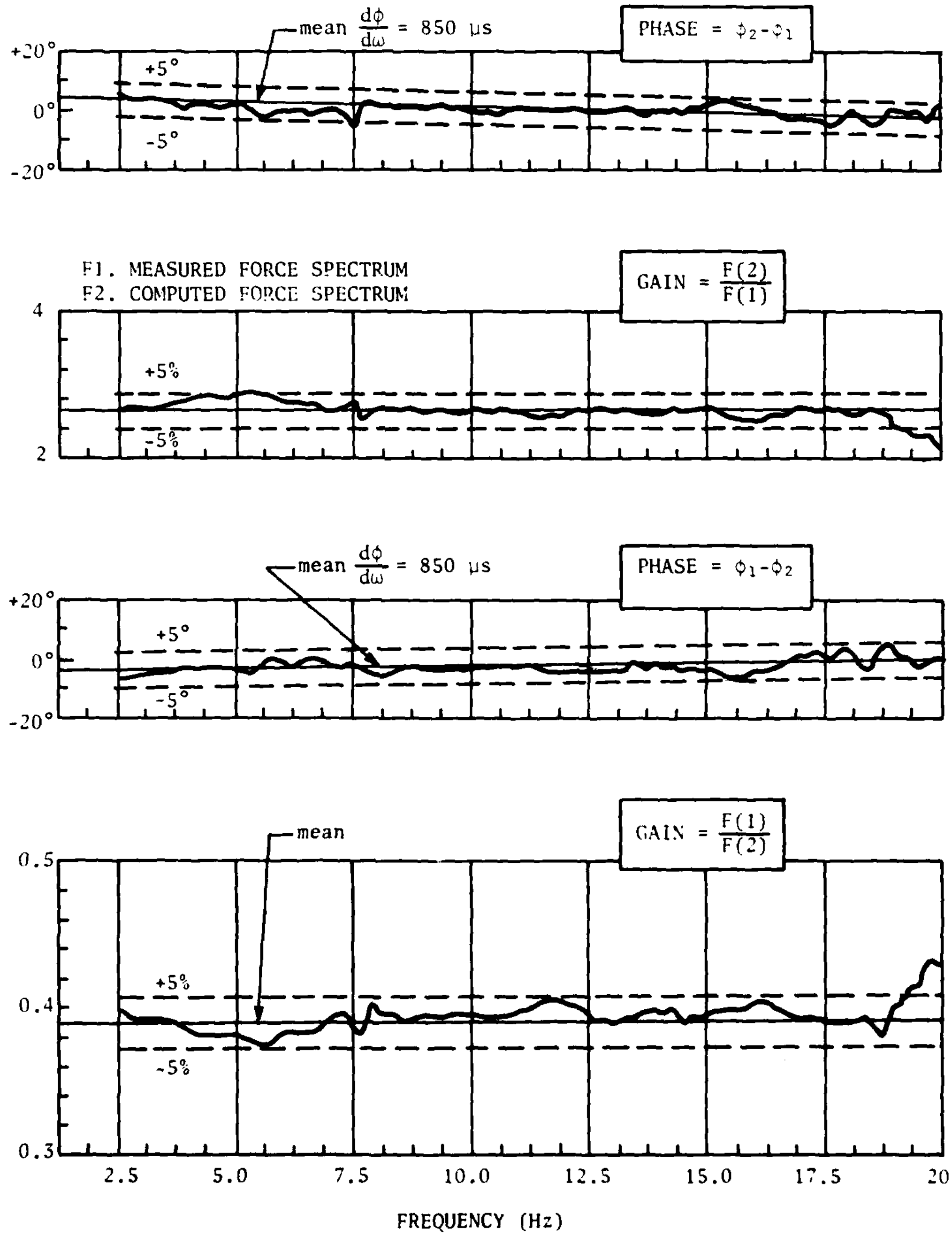


Figure 20. Transfer function

DISTRIBUTION

	Copy No.
EXTERNAL	
In United Kingdom	
Defence Science Representative, London	Cnt Sht Only
Institution of Electrical Engineers	1
British Library, Lending Division	2
In United States of America	
Counsellor, Defence Science, Washington	Cnt Sht Only
Cambridge Scientific Abstracts	3
NASA Scientific and Technical Information Office	4
Engineering Societies Library	5
In Australia	
Department of Defence	
Chief Defence Scientist Deputy Chief Defence Scientist Controller, External Relations, Projects and Analytical Studies Superintendent, Science Programs and Administration	6
Director, Joint Intelligence Organisation (DSTI)	7
Director General, Army Development (NSO), Russell Offices for ABCA Standardisation Officers	
UK ABCA representative, Canberra	8
USA ABCA representative, Canberra	9
Canada ABCA representative, Canberra	10
NZ ABCA representative, Canberra	11
Document Exchange Centre Defence Information Services Branch for:	
Microfilming	12
United Kingdom, Defence Research Information Centre (DRIC)	13 - 14
United States, Defense Technical Information Center	15 - 26
Canada, Director, Scientific Information Services	27

DOCUMENT CONTROL DATA SHEET

Security classification of this page: UNCLASSIFIED

1 DOCUMENT NUMBERS	
AR Number:	AR-004-156
Series Number:	AEL-0242-TM
Other Numbers:	

2 SECURITY CLASSIFICATION	
a. Complete Document:	Unclassified
b. Title in Isolation:	Unclassified
c. Summary in Isolation:	Unclassified

3 TITLE

FLUTTER GENERATOR CONTROL AND FORCE COMPUTER

4 PERSONAL AUTHOR(S):

R.W. Levinge

5 DOCUMENT DATE:

July 1985

6

6.1 TOTAL NUMBER OF PAGES 52

6.2 NUMBER OF REFERENCES:

7

7.1 CORPORATE AUTHOR(S):

Advanced Engineering Laboratory

7.2 DOCUMENT SERIES AND NUMBER

Advanced Engineering Laboratory 0242-TM

8 REFERENCE NUMBERS

a. Task: AIR 82/249

b. Sponsoring Agency: AFRR 3/83

9 COST CODE

10 IMPRINT (Publishing organisation)

Defence Research Centre Salisbury

11 COMPUTER PROGRAM(S) (Title(s) and language(s))

12 RELEASE LIMITATIONS (of the document):

Approved for Public Release

Security classification of this page: UNCLASSIFIED

Security classification of this page: UNCLASSIFIED

13 ANNOUNCEMENT LIMITATIONS (of the information on these pages):

No limitation

14 DESCRIPTORS:

a. EJC Thesaurus Terms

Flutter
Computation
Speed control
External stores
Mathematical models

b. Non-Thesaurus Terms

15 COSATI CODES:

01020

16 SUMMARY OR ABSTRACT:
(if this is security classified, the announcement of this report will be similarly classified)

It is required to investigate the possibility of flutter induced by a store carried under the wing of an aircraft. This involves in-flight dynamic analysis of structural deformations at given points on an airframe due to forces originating in the store. A system of rotating eccentric masses generates a force spectrum 2.4 to 20.0 Hz in both horizontal and vertical axes. Electronically controlled, the "Flutter Generator" runs for 28 s with a swept frequency and a peak force of 800 N. The vertical component of force is computed continuously and telemetered to ground as an analogue signal.

Security classification of this page: UNCLASSIFIED

The official documents produced by the Laboratories of the Defence Research Centre Salisbury are issued in one of five categories: Reports, Technical Reports, Technical Memoranda, Manuals and Specifications. The purpose of the latter two categories is self-evident, with the other three categories being used for the following purposes:

Reports	:	documents prepared for managerial purposes.
Technical Reports	:	records of scientific and technical work of a permanent value intended for other scientists and technologists working in the field.
Technical Memoranda	:	intended primarily for disseminating information within the DSTO. They are usually tentative in nature and reflect the personal views of the author.

END

DATE
FILMED

1-86

DTI